2013

Adaptive Semantic Annotation of Entity and Concept Mentions in Text

Pablo N. Mendes
Wright State University - Main Campus

ADAPTIVE SEMANTIC ANNOTATION OF ENTITY AND CONCEPT MENTIONS IN TEXT

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

By

PABLO N. MENDES
M.Sc., Federal University of Rio de Janeiro, 2005
B.Sc., Federal University of Juiz de Fora, 2003

2013
Wright State University

Wright State University
GRADUATE SCHOOL

June 1, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Pablo N. Mendes ENTITLED Adaptive Semantic Annotation of Entity and Concept Mentions in Text BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Amit P. Sheth, Ph.D.
Thesis Director

Arthur Goshtasby, Ph.D.
Director, Computer Science & Engineering Graduate Program

R. William Ayres, Ph.D.
Interim Dean, Graduate School

Committee on
Final Examination

Amit P. Sheth, Ph.D.

Krishnaprasad Thirunarayan, Ph.D.

Shajoun Wang, Ph.D.

Sören Auer, Ph.D.

ABSTRACT

Mendes, Pablo N. Ph.D., Department of Computer Science and Engineering, Wright State University, 2013. *Adaptive Semantic Annotation of Entity and Concept Mentions in Text.*

The recent years have seen an increase in interest for knowledge repositories that are useful across applications, in contrast to the creation of ad hoc or application-specific databases. These knowledge repositories figure as a central provider of unambiguous identifiers and semantic relationships between entities. As such, these shared entity descriptions serve as a common vocabulary to exchange and organize information in different formats and for different purposes. Therefore, there has been remarkable interest in systems that are able to automatically tag textual documents with identifiers from shared knowledge repositories so that the content in those documents is described in a vocabulary that is unambiguously understood across applications.

Tagging textual documents according to these knowledge bases is a challenging task. It involves recognizing the entities and concepts that have been mentioned in a particular passage and attempting to resolve eventual ambiguity of language in order to choose one of many possible meanings for a phrase. There has been substantial work on recognizing and disambiguating entities for specialized applications, or constrained to limited entity types and particular types of text. In the context of shared knowledge bases, since each application has potentially very different needs, systems must have unprecedented breadth and flexibility to ensure their usefulness across applications. Documents may exhibit different language and discourse characteristics, discuss very diverse topics, or require the focus on parts of the knowledge repository that are inherently harder to disambiguate. In practice, for developers looking for a system to support their use case, is often unclear if an existing solution is applicable, leading those developers to trial-and-error and ad hoc usage of multiple systems in an attempt to achieve their objective.

In this dissertation, I propose a conceptual model that unifies related techniques in

this space under a common multi-dimensional framework that enables the elucidation of strengths and limitations of each technique, supporting developers in their search for a suitable tool for their needs. Moreover, the model serves as the basis for the development of flexible systems that have the ability of supporting document tagging for different use cases. I describe such an implementation, DBpedia Spotlight, along with extensions that we performed to the knowledge base DBpedia to support this implementation. I report evaluations of this tool on several well known data sets, and demonstrate applications to diverse use cases for further validation.

Contents

List of Figures

List of Tables

Acknowledgments

My experience in graduate school was full of adventures. If I were to properly thank every person that helped me along the way, this section would be much longer than the sum of the chapters of the dissertation together. I hope that each and every one of you knows how much it is the case that none of this would be possible without your participation. My victory is, by all means, also yours.

I would like to start by thanking my adviser, Amit Sheth, for his support and encouragement, and for all that I learned from him. Many thanks to my dissertation committee members T.K. Prasad, Shaojun Wang and Sören Auer for all their insightful comments on my research and this dissertation.

I especially thank my labmates Christopher Thomas and Cartic Ramakrishnan for being my mentors and big brothers throughout my studies, and Meena Nagarajan for her sweetness, friendship and professional example. They inspired me personally and professionally, and supported me through the hardest times. Thanks to Karthik Gomadam and Ajith Ranabahu for their comradery and example. To Wenbo Wang, thanks for bringing renewed energy and huge talent, and for your continued friendship and support. To Pavan Kapanipathi, thanks for his patience, understanding, trust and encouragement. Pavan taught me a ton about being a mentor and about being a friend. To all of my Kno.e.sis friends, thanks for helping me to get back up every time that I fell. To the Brazilian community in Dayton: Pablo, Percio, Star, Sandro, Leandro, Rafael, Marcos, Erik, Felipe, Dani, Ana, Debora... you helped make it all so much more fun!

Thanks to Alberto Dávila, Maria Luiza M. Campos and Ligia Barros for their advisement during my master's, for encouraging me to join a PhD program, and for introducing me to Jessica Kissinger. I thank Jessie for offering me a scholarship and for warmly welcoming me to the USA. Thanks also to Kathy Couch, Michael Luchtan, Mark Heiges, Chih-Horng Kuo, Adriana de Oliveira, Abijeet Bakre and many other friends for their patience and support in my first years in a completely new world. Thanks to Sadler and the Crofts for

all they taught me, and for being such a beautiful family. To the Brazilian community in Athens: Anita, Mara, Fernanda, Cissa, Maria Gloria (yes, you're one of us), Thomas, Felipe, Raphael, Lara... you rock!!!

Thanks to my friends Chris Eckenroth and Robert Moser for all of their support through the tough times of career change when I decided to join Dr. Sheth in the creation of the Kno.e.sis Center at Wright State. To Lee Pratt and Marie Michelle Cordonnier-Pratt, who took me in at the Lab for Genomics and Bioinformatics, and made me feel like a family member: I am forever thankful for their generosity.

During my stay in Germany I made new friends that also supported, inspired and helped me grow tremendously. Thanks to Chris Bizer for receiving me in Berlin with open arms, for his guidance, and for entrusting me with the leadership roles that I took while working in his group. Thanks to Max Jakob for being an amazing friend and collaborator. Max implemented some of the key DBpedia extractors that were used in the original DBpedia Spotlight paper, and has given great feedback in several papers since. Thanks to Anja Jentzsch for being always available to help and for her generous contributions to the WBSG. Thanks to Oscar Corcho for his guidance and insightful comments on an early draft of this dissertation. I specially thank Sören Auer for stepping in during difficult times and for providing selfless help and advice. Thanks to Andreas Schultz, Robert Isele and Christian Becker for all the great work that set the stage for my contributions with Sieve. Thanks to Jo Daiber and Dirk Weissenborn for bringing renewed energy and great talent to the DBpedia Spotlight open source project. Their contributions have picked up where I left, and are taking our project to places I would not manage to take on my own.

Lastly, many thanks to my family. Thanks to my loving partner Orsolya Szabó. I have no words to properly describe my gratitude for all the sacrifices she has made on my behalf, for her encouragement at the hardest times, and for her endless patience throughout these years. To Bea and Sándor, thanks for making me feel welcome, appreciated and loved, even though I was so far from home. Special thanks to Nagyi Irene for her affectionate

care, delicious food and skillful mastery of my native language!

I am forever indebted to my mom and dad, for their understanding and unconditional support for all these years. I thank them for being always present even when I was physically distant. Thanks for all they did for me, for being a lifelong example, and for being such amazing friends! Thanks to my brother and sister Vinicius and Polyanna. And thanks to my extended family Rose, Aline e Paloma.

To my grandmother Therezinha, my appreciation for caring for me also as a mother and enabling the start of this career when I was still 15 years old. I am sorry it took me so long to get here. This dissertation is dedicated to you.

Dedicated to my grandmother.

À minha avó Therezinha.

Introduction

1.1 Historical Context

The Web has developed into the most important knowledge repository ever built. Information on the Web supports a wide variety of tasks, from routinely finding public transportation timetables, to performing cutting-edge research that relies on complex results from scientific experiments. Nowadays, virtually any information seeking activity includes the use of information sources shared through the Web.

Most of the knowledge on the Web is exposed in the form of documents written in natural language. Although natural language is structured by grammatical rules, it is often regarded as *unstructured data* as its structure is implicit – i.e. the grammatical rules are not explicitly encoded in the text. Contrast this to relational databases, where data is always shared in n-tuples following a pre-defined and explicit schema.

A more recent movement has been exponentially increasing the amount of structured data available on the Web – data where the structure is explicitly encoded. The Linking Open Data project is a community effort to publish and interconnect Open Data sets according to the Linked Data principles set forth by the Web's inventor Tim Berners-Lee[1]. Linked Data sources use URIs to globally identify things, HTTP as a protocol for information requests, and Resource Description Framework (RDF) [Lassila and Swick, 1999] to encode the information being exchanged. RDF structures information in subject-predicate-

[1] http://www.w3.org/DesignIssues/LinkedData.html

object (SPO) triples that allow one to describe a series attributes of a subject, as well as relationships between subjects. Linked Data sources include facts from several domains, such as sports, media and entertainment, and have become a standard for many governments and their Open Data initiatives.

```
:Rio_de_Janeiro :areaTotalSqMi "486.500000"  .
:Rio_de_Janeiro :populationTotal "6,323,037" .
:Rio_de_Janeiro :country :Brazil .
```

Figure 1.1: Example SPO triples describing the Brazilian city of Rio de Janeiro (identified by the URI `Rio_de_Janeiro`). It describes Rio's total area in square miles through a property (`areaTotalSqMi`) and relates it to another resource (`Brazil`) through the property `country`. Namespaces have been omitted for readability.

Many of these data sources serve as reference databases for defining entities and relationships between entities. They can serve as hubs to interconnect other data sources, including structured and unstructured data [Immaneni and Thirunarayan, 2007, Thirunarayan and Immaneni, 2009]. The combination of structured and unstructured data using the Web as a platform allows both Web-style browsing, as well as database-style queries that span both the structured and the unstructured worlds. This enables uncountable opportunities for remixing and repurposing data, considering the amount of textual and structured information currently available.

The task of scanning unstructured data and tagging mentions of entities described in structured data is a prerequisite bridging these two worlds. A number of initiatives have turned to the problem of recognizing names and phrases in text and associating them with unambiguous entity and concept identifiers from a knowledge base. Techniques useful for this task have been approached in literature from different angles, and under different task names such as keyphrase extraction, topic indexing, named entity recognition and disambiguation, word sense disambiguation, entity linking, entity extraction, concept tagging, etc. We present formal definitions in Chapters 2 and 3, and discuss similarities and distinc-

tions between each of those tasks.

Most of previous work has focused on techniques and systems specializing in some of those tasks, in particular settings, with particular text genres, entity types, or for a specific application. Due to the wide spectrum of applications that become possible through the integration of structured and unstructured information, we argue for the need to generalize across these tasks. This becomes particularly important with the growing importance of Cloud Computing, as more applications rely on platform and data as a service. In the context of an emerging Annotation as a Service (AaaS) paradigm, systems that are able to adapt to user-specific needs at run time reduce costs and increase usefulness.

In the remainder of this chapter, we present a number of motivating applications, summarize our objectives and contributions, and describe the structure of this dissertation.

1.2 Use Cases for Semantic Annotation

In contrast to the early uses of natural language processing in defense and in specialized in-house applications, the Web is an interactive medium where the information supply is much more user driven. There are different user groups and end users of information systems with very different needs. There is a growing need for services that developers can use to enrich content to support contextualized interactions or more expressive organization of content. Consider, for instance, the following applications.

News. News companies such as The New York Times and The Guardian are examples of companies that have recognized the usefulness of the interconnection of text and a knowledge base. Many news websites already publish annotations of 'topics' within their online news articles. These topics are marked in the Web page of the article, and linked to topic pages that serve as hub for articles mentioning those topics, while also linking to related topics and other relevant articles. Those annotations focus on named entities such as Peo-

ple, Locations and Organizations, and especially on prominent ones. For instance, mentions of the president of a country (a prominent person) are annotated, but mentions to an interviewed bystander are not. On the one hand, adding more links to an article provides more ways for users to interact with the content and stay within a particular news site. On the other hand, too many links may be distracting or aesthetically displeasing. If a computer program is tasked with automatically reproducing editorial tags, there are a number of features besides correctness of annotation that may be important for satisfactory results. These may include the fact that certain entities are more important, or that certain mentions are not relevant to the context. An understanding of how these features can be enabled by KBT systems is very important for these applications.

Educational Material. Coursework, training material and educational text in general have the focus to discuss concepts and their inter-relations, often involving concepts that are unknown to the reader. The annotation of such text can help guide the reader's exploration, allowing one to retrieve explanations for unknown concepts while reading the text. In this use case, there is no focus on particular types, and more abstract concepts such as Fire or Emergency become important, while in the News Articles use case they may be considered trivial and therefore not annotated.

Semantic Search. There are more than 20 different cities called Berlin. State-of-the-art Web search technologies commonly rely on popularity measures to rank the most likely pages referring to a given keyword set. One of the assumptions behind Semantic Search is that such techniques can be problematic in the long-tail of less popular entities, and that semantic identification of concepts could improve results in such cases. For this use case, the identification of both Named Entities and more general concepts is of major interest. This is due to the fact that, for instance, some concepts can also be confused with named entities (as in '*huge*' versus `Huge_(album)`), and searches for one or another should be possible. Moreover, differently from the News use case, it is key to annotate as many

entities and concepts as possible, so that a comprehensive index of the Web can be built.

Social Media. With the creation of discussion forums and, more recently, the explosion microblogging platforms, a number of researchers have turned to those sources of conversation as way to gain an understanding of their target public's ideas, feelings and behavior. In contrast with the News and Educational Material use cases, at least two classes of problems stand out when dealing with social media text from forums and microblogging: the informal nature of the text, and the volume and speed at which this information is generated. First, informal text often exhibit unorthodox grammar and misspellings, besides including higher levels of name variation for entities (e.g. use of slang to refer to things). Second, with a high volume of information arriving at a high speed, it quickly becomes necessary to develop ways to automatically contextualize the information so as to alleviate the problem of information overload.

Previous work has focused on specialized solutions for each use case, or even specializations of some of the use cases presented above. Our hypothesis is that we can conceptually model and implement a generic core solution that can be adapted across-use cases.

1.3 Thesis statement

This dissertation analyzes the problem of adaptable Knowledge Base Tagging (KBT) and shows that a solution is feasible and useful for many applications. The KBT task overlaps with several currently studied information extraction tasks, but goes beyond each one of them individually in that a combination of (parts of) each task may be needed to accomplish a user's needs. A novel conceptual model that encompasses the similarities and differences between relevant information extraction tasks is proposed. This model offers two contributions: i) enables cross-task system evaluations, ii) provides the needed flexibility for performing custom information extraction tasks defined at runtime. In connection

with the Knowledge Base (KB), this model empowers the users of AaaS systems to provide much richer annotation task specifications that include information from the KB. This work is the first to study KBT as a service. This setting differs from the setting assumed for most machine learning-based annotation approaches, since training data describing the needs of each particular annotation use case may be limited or non-existent.

1.4 Organization

Chapter 2 describes background and related work. Chapter 3 introduces the conceptual model. Chapter 4 describes the knowledge base that will be used as the focus of our implementation. Chapter 6 presents the core evaluations. Chapter 7 discusses applications of our approach to a number of distinct domains. Chapter 8 makes final considerations and discusses possible future extensions.

1.5 Notation

Throughout this dissertation, we will adopt a notation scheme as follows. We use capital letters for sets or vectors and lowercase letters with a subscript index for individual items in these sets or vectors. For example, the sentence 'The book is on the table' can be represented by the vector of words $S = \{w_1, ..., w_6\}$, where $w_1 =$ The, $w_2 =$ book, ..., $w_6 =$ table. Mentions to the components of our proposed conceptual model are formatted with sans-serif font, e.g. Annotation. Diagrams, when possible, are displayed in Unified Modeling Language (UML), summarizing interactions, sequences, activities, etc.

Background and Related Work

Several information extraction tasks that overlap with KBT have been proposed in the past. They differ on the context in which they were created and on their primary objectives. The aim of this chapter is to introduce the reader to these tasks, identify overlaps, spell out the most important distinctions, and lay out the foundation for developing a framework that will allow cross-task applications and evaluations.

We start by describing the methodology used for the literature review conducted for this dissertation (Section 2.1), followed by a discussion of the relevant background (Section 2.2) and a presentation of the state of the art (Section 2.3). In Section 2.4 we present other cross-task models that relate to the KBT task. Lastly, Section 2.5 concludes the chapter with a summary of the literature review performed.

2.1 Methodology for the Literature Review

Our literature review was performed according to the following methodology: (i) select landmark papers, (ii) select key terms, (iii) select conferences and journals, and (iv) review citations to landmark papers and other papers containing key terms from selected fora.

We started by selecting landmark papers and reviewing citations to the landmark papers up to the year 2012, as well as papers cited by the landmark papers. The list of landmark papers used includes Hammond, Sheth, and Kochut [2002], Dill et al. [2003], Bunescu and Pasca [2006], Cucerzan [2007], Mihalcea and Csomai [2007], Milne and

Witten [2008], Kulkarni et al. [2009].

Based on the review of the discussion in landmark papers, the second step was to compile a list of key terms that we consider central to the topic. The list of terms includes:

> CONCEPT TAGGING, ENTITY EXTRACTION, ENTITY DISAMBIGUATION, ENTITY LINKING, ENTITY TAGGING, NAMED ENTITY RECOGNITION, PHRASE RECOGNITION, PHRASE SPOTTING, SEMANTIC ANNOTATION, SEMANTIC MARKUP, TAG EXTRACTION, TOPIC INDEXING, WORD SENSE DISAMBIGUATION, WORD SPOTTING

Figure 2.1: Key terms used in the literature review.

The third step was to select primary and secondary venues, and lastly to review papers in the past 5 years in those venues that contain one of the selected terms in the title. For each of the matches, we pre-screened the abstract, and proceeded with a full review of the paper in case the subject of the paper was considered relevant to the research discussed in this dissertation. We primarily focused on Semantic Web, Computational Linguistics and Natural Language Processing fora. However, relevant papers from related areas are also occasionally added, including related research in areas such as Knowledge Discovery, Data Mining, Information Retrieval and Database Systems.

Semantic Web and Web Engineering	WWW, CIKM, ISWC, WI, ESWC, JWS, SWJ, ISJWIS, WISE, I-SEMANTICS
Comp. Linguistics and NLP	ACL, EACL, NAACL, SemEval, EMNLP, COLING, HLT, IJCNLP, LREC, RANLP
Other (DB, IR, data mining)	KDD, WSDM, SIGIR, SIGMOD, VLDB

Table 2.1: Scientific forums chosen as focus for the literature review.

2.2 A Review of Information Extraction Tasks

This section presents a review of tasks within the field of information extraction that are most relevant to KBT.

Automatic Term Recognition Studies of Automatic Term Recognition (ATR) [Kageura and Umino, 1996] start from the realization that there is a vast amount of textual content that use language that is so specialized that they form a kind of sublanguage. Examples are technical documents, or documents with focus on a particular domain. The objective of ATR is to discover phrases in these 'sublanguages' that are significant for a given domain – i.e. terms – usually with the aim of building a conceptual system organizing these terms – i.e. a terminology.

Techniques for ATR vary from models of word and term structures motivated from lexical morphology [Ananiadou, 1994] to corpus-based statistics [Kageura and Umino, 1996]. In relation to KBT, ATR can be seen as a technique for discovering new terms to be added to a KB. When a ATR system discovers a term that already exists in the KB, it can be seen as a form of entity or concept name recognition. ATR does not, however, rely on the notion of a reference terminology that is used during the process, as its objective is to create one such terminology based on corpora.

Keyphrase Extraction Another task, with objectives overlapping those of ATR, is called Keyphrase Extraction (KE) [Frank et al., 1999]. It relies on notions of significance, dubbed keyphraseness, to establish which segments of text to extract from a document or collection of documents. The significance of a phrase can be established as defined by the user, or as classified by how well phrases capture the contents of the input text. When significance is defined with regard to a sublanguage pertaining to a domain of knowledge, KE is very similar to ATR. Studies of KE often break down keyphraseness in two notions: phraseness and informativeness.

Phraseness aims to distinguish sequences of words that form a meaningful phrase – according to some definition of meaningful. For instance, while "is the" is a two-word collocation that carries little meaning, the words "What's up" form a phrase that can be used as a greeting in American street language or as the name of a song by 4 Non Blondes

– among at least 4 possible meanings described in Wikipedia. There is no unique definition for what should be considered a phrase. For instance, while some users will be concerned with noun phrases, others may be focusing on actions, or specific types of entities. The most common approaches for estimating phraseness focus on obtaining statistical measures of word association strength from corpora. Various metrics have been used for analyzing word collocations, including the chi-squared test and the pointwise mutual information (PMI) [Church and Hanks, 1990], the t-test [Church et al., 1991], the binomial log-likelihood ratio (BLRT) [Dunning, 1993] and the mean and variance [Smadja, 1993]. Pantel and Lin [2001] combines PMI and BLRT, while Tomokiyo and Hurst [2003] propose a language-model approach that models phraseness as the loss in information by applying a unigram model instead of a n-gram model.

Informativeness aims at modeling how well a phrase captures the contents of a subject. Definitions of 'subject' vary from the particular discussion in an article, to the general topics discussed within a certain domain of knowledge. Informativeness is often estimated from analyzing phrase occurrences in a target corpus, or by analyzing the differences in occurrences in two corpora [Damerau, 1993][Tomokiyo and Hurst, 2003].

Named Entity Recognition Named Entity Recognition (NER) is defined as the task of inserting "tags into the text to mark each string that represents a person, organization, or location name, or a date or time stamp, or a currency or percentage figure" [Grishman and Sundheim, 1996]. Nadeau and Sekine [2007] present a survey of NER techniques including supervised, semi-supervised and unsupervised techniques. Typical features used by NER systems include word-level features (such as capitalization, pre-/suffixes, punctuation, etc.), list lookup features (gazetteers, lexicons or dictionaries), as well as corpus-level features (multiple occurrences, syntax, frequency, etc.). The most effective approaches rely on supervised machine learning of sequence labels with techniques such as Conditional Random Fields (CRF)s.

In general, like in KE, in a NER task there is no explicit notion of a target knowledge base, or concept identifiers. The objective of KE is to build a set of terms in a bottom-up fashion in an application or corpus-specific manner. The objective of NER is to detect and segment mentions of entities of certain types. There have been, however, approaches that attempt to leverage a set of entities (e.g from Wikipedia) as providing a controlled vocabulary used as a lexicon to influence NER systems [Ritter et al., 2011]. Still, in their approach, the KB figures as a mere provider of dictionaries.

Differently from ATR, KE, and NER, which focus on recognizing phrases with particular characteristics in text, other information extraction tasks have focused on linking text to a KB identifiers. Since entries in KBs may have many different names, this requires resolving ambiguity – i.e. disambiguating.

Word Sense Disambiguation Word Sense Disambiguation (WSD) can be defined as the the task of automatically identifying the meanings of words in context [Agirre and Edmonds, 2006, Navigli, 2009]. The problem has been approached in two settings. All-words WSD targets the association of a unique sense identifier to *each and every word* in the input text. Targeted WSD uses the input text to determine the sense of *one* given ambiguous word [Stevenson and Wilks, 2003]. Since the segmentation of the input words is already given, selecting phrases to disambiguate is not a challenge discussed in the WSD literature.

The most common sense repository in the WSD literature is WordNet [Miller, 1995]. WordNet can be seen as a sort of general purpose knowledge base. However, it focuses on the meaning of more general words, and contains only a limited set of relationships between these concepts.

Co-reference Resolution The objective of Co-reference Resolution (CR) is to determine which mentions in text refer to the same entity. Entities can be mentioned by name (e.g. Barack Obama, Mr. Obama), nominally (the president of the USA), or pronominally (he,

him). CR's aim is then to cluster the mentions that refer to the same entity – in this case, he, Barack Obama, is the president of the USA. Traditionally, CR focuses on co-referent mentions in the same document or across documents (Cross-document Co-reference Resolution). Connecting these mentions to a KB is not within the focus of CR literature. However, some researchers have used relatedness features mined from Wikipedia [Ponzetto and Strube, 2006] and from KBs [Zheng et al., 2013, Wick et al., 2009] to help with the CR task. Moreover, in formulations of CR where the KB is present, the task can be performed jointly and in synergy with the task of linking co-referents to a knowledge base [Zheng et al., 2013].

Entity Linking. Entity Linking (EL) is known as the task of associating entity mentions recognized in text with their corresponding identifiers in a KB. The input of the task is an entity name and an example document where that name was used. The output is an entity identifier for the referent of that mention in the KB. Equivalent problems to EL have been mentioned in the literature, including Named Entity Disambiguation (NED) [Cucerzan, 2007], and Entity Resolution [Pereira et al., 2011].

Although the main challenge addressed in EL is ambiguity, another important challenge in EL is dubbed the NIL problem. Some mentions refer to entities that are not present in the KB. To complicate matters, some mentions are homonyms to entities in the KB, but refer to an out-of-KB entity. More recently, the EL research community has proposed the clustering of NIL entities, further narrowing the gap between EL and CR.

Wikification Wikification refers to the task of automatically marking up phrases with links to Wikipedia pages that describe those phrases. It involves both recognition and disambiguation challenges. It needs to recognize which of the words mentioned in the refer to an entity, while also determining which specific entity was the referent – in case more than one interpretation is possible.

For the recognition challenge, previous work [Mihalcea and Csomai, 2007, Milne

and Witten, 2008] has adopted Wikipedia's definition of keyphraseness. In Wikipedia's manual of style[1], only the first mention of notable entities in an article should be annotated, and there is a focus on non-obvious references. If a wikilink would not contribute to the understanding of a specific article, the Wikipedia Manual of Style would discourage its creation.

For the disambiguation challenge, different sources of context and modeling techniques have been proposed, spanning from bags-of-words for each article, all the way to graph-based measures relying on links between Wikipedia articles. Disambiguation is at the core of KBT and will be studied in more detail in what follows.

With relation to KBT, Wikification can be seen as a way to tag textual documents to a KB where entities are identified by Wikipedia URLs. However, it differs in the sense that there is no notion of entity type in Wikipedia[2]. As a result, Wikification systems usually do not focus on particular types, simply linking text to identifiers drawn from the set of Wikipedia page URIs.

Text Normalization. Text Normalization refers to a process of transforming free-form text input into a canonical form by eliminating 'noise' in the text. In the context of Information Retrieval (IR), when someone searches for a word (e.g. 'run'), it is useful to return also documents containing different forms of the same word ('*running*', '*ran*', '*runs*'). This kind of word normalization is usually performed through stemming and lemmatization. Text normalization is also frequently used for mapping acronyms, abbreviations, numbers and dates to a canonical form, in order to ease post processing such as database look-ups or text to speech conversion. Similarly, normalization is also applied to entities and concepts, as there are several ways one can refer to the same entity (e.g. 'George H. Bush', 'Bush senior', etc.). Therefore, the problem of reducing entity name variations into a canonical form relates to CR and Entity Linking problems. Moreover, one can see KBT

[1] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)

[2] although there is substantial work in how to automatically detect entity types from Wikipedia's category system, infoboxes, abstracts, etc.

as performing text normalization in the presence of domain knowledge.

Topic Indexing. Topic Indexing (TI) originates from the need to organize a large body of cultural goods – e.g. books in a library – for subsequent retrieval. It usually relies on a manually constructed set of terms that serves as a controlled vocabulary for describing those cultural goods. In TI, the aim is to identify for each item – e.g. a document – the topics that are the most significant for describing that item's content (Maron, 1977). It relates to Wikification, if Wikipedia is used as the controlled vocabulary used to index the main topics in a document [Medelyan et al., 2009]. It also relates to KE in the sense that topic indexers rely on the notion of aboutness.

More recently, terms like Tag Extraction and Entity Extraction have been used to describe similar tasks to TI. Like Keyphrase Extraction, entities are extracted on document level rather than on mention level (there is no notion of position of the extracted reference). Unlike KE, it focuses on a pre-established set of entities.

Semantic Annotation. A related task has been discussed in the Semantic Web community and regarded as Semantic Annotation. It often refers to tagging text with references to an ontology defining terms from a domain. This task is the closest to the KBT task we propose. We avoid the term Semantic Annotation to prevent confusion with other tasks in Computational Linguistics that can also be construed as semantic annotation, such as semantic role labeling. Moreover, beyond the traditional definition of semantic annotation, KBT also considers aspects of annotation that make a correct annotation fit or not fit for use in an application (e.g. general concepts should not be annotated, or only the most relevant concepts should be annotated, etc.). Meanwhile Semantic Annotation as studied in the Semantic Web community so far has focused on correctness of sense disambiguation.

An early proposal in this space, the Semantic Enhancement Engine (SEE) [Hammond et al., 2002] relied on a modular architecture to extract entity mentions and annotate them with identifiers from a knowledge base. Their modules included string-matching to extract

entity names, regular expression-based extraction, domain classification and the use of relationships between entities in the knowledge base to try to narrow down the number of possible matches for each name. To the extent of our knowledge, evaluations of their entity tagging techniques were not presented.

Other tasks. There are a number of other IE tasks that also relate to KBT to a lesser extent, as they either can provide useful features or can use KBT features in order to perform better. Those include Relationship Extraction, Summarization, Machine Translation, etc. The review of those techniques falls out of the scope of this dissertation. The task of Identity Resolution [Elmagarmid et al., 2007] – also known as Entity Resolution, Deduplication, Duplicate Record Detection, or Record Linkage – is widely discussed in the Database Systems research community. In that task, the objective is to interconnect two databases or knowledge bases, identifying which records in each database are co-referent. It is formulated as a N to N problem – mapping N records in one database to their corresponding N records in the other database. The number of records (N) is usually very large. Meanwhile, Entity Linking in the context of NLP is usually construed as a 1 to N (local disambiguation) or a C to N problem (global disambiguation) – where C is the number of entities in a document, a number that is closer to a small constant than to N, the number of entities in a KB. Therefore, a large portion of the research in Identity Resolution is dedicated to efficient ways to perform these N by N comparisons between database records. The study of Entity Resolution in the context of Database Systems is out of the scope of this dissertation.

Objective	ATR	KE	NER	EL	WSD	WKF	KBT	TI	CR
Recognize new terms (Phrase Discovery)	x	x	x	x (NIL)			x		
Classify ontological type (Phrase Classification)			x	x			x		
Phrase Recognition							x		
Resolve ambiguity				x	x	x	x	x	x
Measure importance/relevance	x (to domain)	x (to text)					x	x	
Produce annotations for consumption						x	x	x	
Positional			x			x	x		x
Reference Knowledge Base									x (document)

Table 2.2:

2.3 State of the art

Most of previous work focuses on the annotation of Named Entities. ATR and KE research have attempted the identification of more general concepts, but the focus is not to disambiguate. WSD research has approached the disambiguation of general concepts in both focused and all-words settings, but to the extent of our knowledge, the combination of keyphraseness and WSD for the user-adaptive annotation of concepts has not been fully explored.

Many approaches for entity annotation have focused on annotating salient entity references, commonly only entities of specific types (Person, Organization, Location) [Hassell et al., 2006, Rowe, 2009, Volz et al., 2007, Gruhl et al., 2009] or entities that are in the subject of sentences [Fader et al., 2009]. Hassell et al. [2006] exploit the structure of a call for papers corpus for relation extraction and subsequent disambiguation of academic researchers. Rowe [2009] concentrates on disambiguating person names with social graphs, while Volz et al. [2007] present a disambiguation algorithm for the geographic domain that is based on popularity scores and textual patterns. Gruhl et al. [2009] also constrain their annotation efforts to cultural entities in a specific domain. Our objective is to be able to annotate entities of unconstrained types from a large knowledge base – we evaluate results with the annotation of any of 3.5M entities of 320 types in DBpedia.

Other approaches have attempted the non-type-specific annotation of entities. However, several optimize their approaches for precision, leaving little flexibility for users with application needs where recall is important, or they have not evaluated the applicability of their approaches with more general use cases [Dill et al., 2003, Bunescu and Pasca, 2006, Cucerzan, 2007, Mihalcea and Csomai, 2007].

SemTag [Dill et al., 2003] was the first Web-scale named entity disambiguation system. They used metadata associated with each entity in an entity catalog derived from TAP [Guha and McCool, 2003] as context for disambiguation. SemTag specialized in precision at the cost of recall, producing an average of less than two annotations per page.

Although the conceptual model presented in this dissertation is generic, for the sake of explanation, practical examples and evaluations in this dissertation focus on DBpedia URIs and Wikipedia text. Bunescu and Pasca [2006], Cucerzan [2007], Wikify! [Mihalcea and Csomai, 2007] and Milne and Witten [2008] (M&W) also used text from Wikipedia in order to learn how to annotate. Our work, therefore, builds upon their findings.

Bunescu and Pasca [2006] only evaluate articles under the "people by occupation" category, while Cucerzan 's and Wikify!'s conservative spotting only annotate 4.5% and 6% of all tokens in the input text, respectively. In Wikify!, this spotting yields surface forms with low ambiguity for which even a random disambiguator achieves an F_1 score of 0.6.

Fader et al. [2009] chooses the candidate with the highest prior probability unless the contextual evidence is higher than a threshold. In their dataset 27.94% of the surface forms are unambiguous and 46.53% of the ambiguous ones can be correctly disambiguated by just choosing the default sense (according to our index).

Kulkarni et al. [2009] attempts the joint optimization of all spotted surface forms in order to realize the collective annotation of entities. The inference problem formulated by the authors is NP-hard, leading to their proposition of a Linear Programming and a Hill-climbing approach for optimization. We propose instead a simple, flexible approach that can be easily configured and adapted to task-specific needs, facilitated by the DBpedia Ontology and configuration parameters.

2.4 Cross-task Benchmarks

Although there is much overlap between the tasks discussed in our literature review, no single one encompasses all requirements needed for the use cases described in Chapter 1. Due to their differences in focus and evaluation settings, it is difficult to contrast the applicability of each of the techniques in a cross-task setting. One of the objectives of this

dissertation is to lay out a framework to support conclusive comparisons.

Literature involving comparisons of tools across different information extraction tasks that are the most relevant for KBT include Hachey et al. [2013] and Cornolti et al. [2013]. [Hachey et al., 2013] evaluate entity linking. They reimplement three seminal approaches [Bunescu and Pasca, 2006, Cucerzan, 2007, Mihalcea and Csomai, 2007] and test them on 5 of the most used datasets. They found out that although most of the literature has been dedicated to disambiguation methods, a large portion of errors is actually due to errors in the search for candidate senses for each phrase. This highlights the need for a framework like the one presented in this dissertation, for understanding each step and isolating points where errors can occur. We will present a set of comprehensive evaluation experiments to demonstrate this contribution.

Cornolti et al. [2013] also attempted a comparison of systems that span more than one of the traditional IE tasks surveyed in this section. They divide the approaches according to the ability to Disambiguate to Wikipedia (D2W), Annotate to Wikipedia (A2W), Scored-annotate to Wikipedia (Sa2W), Concepts to Wikipedia (C2W), Scored concepts to Wikipedia (Sc2W), Ranked-concepts to Wikipedia (Rc2W). They evaluate 7 of the best known entity annotation systems on 5 of the most well-known datasets available. In comparison to previous evaluations, this work has provided important contributions in recognizing similarities and distinctions between different approaches. They have swept the configuration spaces in order to obtain different annotation styles, and applied clever two-side measures to account for fuzzy matches. They, however, treat each system as a black box, therefore falling short of identifying reasons why some systems perform better than others.

In general, the following problems exist when performing cross-task evaluations of techniques:

- There is not a commonly accepted way of counting correct annotations.
- External tools are applied off-the-shelf and as a black-box.

- Not every annotation in a gold standard is equally as difficult for systems to obtain.

As one of the contributions of this dissertation, we present a framework to alleviate these problems. The framework presented here offers more encompassing definitions of tasks, more fine grained modeling of tasks and a model of the difficulty to disambiguate mentions. Thus, it provides more robust ways to compare and contrast techniques and assess their ability to adapt to different use cases.

2.5 Conclusion

The process of automatically producing semantic annotations usually comprises two important tasks: recognition and resolution. Recognition – also referred to as Spotting – refers to distinguishing segments of text that should not be tagged from those that should, based on task definitions, e.g. applicable types, relevance or importance. Resolution refers to finding suitable unambiguous identifiers to describe the meaning of those phrases – herein called disambiguation. In its broader sense, KBT does not provide a canned definition of which phrases constitute targets of annotations: all words, only phrases with particular characteristics, only names of entities of certain types, etc. In order to fulfill requirements of use cases listed in Section 1.2, a great deal of flexibility is needed by KBT systems wherein techniques from each of those research areas may be useful. However, contrasting the suitability of techniques created for different purposes is challenging without a unifying conceptual model. This is the objective of Chapter 3.

A Conceptual Framework for Semantic Annotation

In this chapter we describe a conceptual framework for the KBT task. This conceptual framework will serve as a basis to explain the techniques developed as part of this work, as well as to compare approaches with one another. We start by defining in Section 3.1 each of the components of our framework, including the input (text and knowledge base), the output (annotations) and the actors that operate on the input and output (creator, editor, system). We conclude by describing how the proposed framework enables a multidimensional comparison of multiple information extraction tasks, as well as the evaluation of cross-task tools for the fulfillment of user needs in an AaaS setting.

3.1 Definitions

In general, an annotation task can be summarized by the workflow illustrated in Figure 3.1. An agent (typically a person) creates a document. This document is sent through a system that performs an automated analysis of the content, detecting if it makes references to entities or concepts that belong to a knowledge base of interest. The annotated document may undergo moderation by an agent (typically a person, but potentially a system) performing the role of editor. Editors may correct wrong annotations, add missing references or re-

move annotations that are deemed uninteresting or unfit for the intended use. Finally, the annotated document is employed for a given objective by an agent playing the role of consumer. Consumers may evaluate the fitness for use of the annotated document, considering the number of errors, annotations out of scope, etc. Unfit documents may be sent back a number of times through any of the previous steps, until it is considered fit.

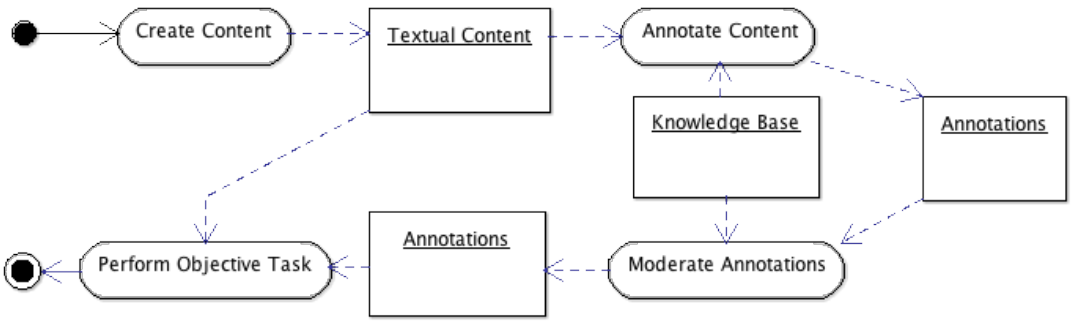


Figure 3.1: Workflow of a KBT task.

The roles of creator, editor and consumer may be performed by distinct agents, or the same agent may perform more than one role. It is also possible to see the Annotate Content activity as being performed by a person (manual annotation), but since our focus is on automatic annotation, we always view this role as one performed by a system. Figure 3.2 depicts the use cases involved in the annotation workflow. More details on the workflow are provided in the remainder of this section.

3.1.1 Actors

In a typical KBT task, actors may perform the roles of Creator, Annotator, Editor or Consumer. These roles may be played by distinct actors, or the same actor may play more than one role. Moreover, each of these roles may be played by human actors, in a 'manual' annotation setting, or by systems, in an automatic annotation setting. Figure 3.2 summa-

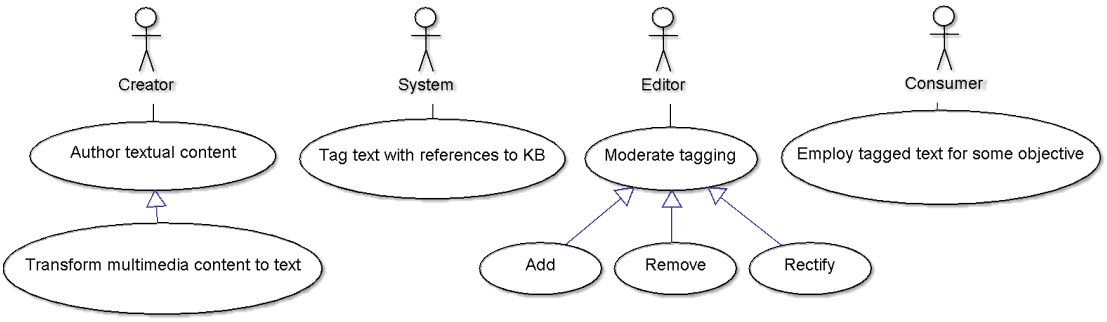


Figure 3.2: Interactions between actors and a KBT system.

rizes actor interactions in a UML Use Case Diagram. This section provides definitions for each actor.

Definition 1. The Creator is the person or system that produced the Textual Content being analyzed by the system.

In some annotation tasks the profile of the content creator might be available, containing descriptions of interests, or other characteristics that help with interpreting the Textual Content created. Therefore it is important to explicitly model the Creator.

Definition 2. The Annotator is the person or system that produces Annotations associating the Textual Content with the Knowledge Base being used by the system.

In some use cases, the annotation is performed by a human expert. We will focus on use cases where the Annotator role is performed by an automated system.

In an AaaS scenario, systems are often operated by users with varying levels of knowledge of the information extraction task being performed. Often, the user calling the system is an expert involved in deciding which of the automatically suggested tags should be kept. In other cases, the user is knowledgeable in the general area of KBT – i.e. a 'power-user' – but not an expert on the content being discussed in the documents.

Definition 3. The Editor is the person or system that moderates Annotations and judging

if they should be kept or discarded following a set of guidelines derived for a specific Objective.

Editors can be seen as a type of Annotator, as each edit is equivalent to the creation of a new annotation, or a deletion of a previously created one, or both. For use cases where a person is performing the role of Editor, there is an inherent trade off with respect to the amount of information that is shown. On the one hand, people are known to favor "recognition over recall" [Hertzum and Frøkjær, 1996] – in other words, displaying many tags could help users to recognize concepts that they would otherwise have not thought of annotating. On the other hand, displaying too many tags can overburden the editor and lower their overall productivity. Adaptive systems should support a flexible definition of how many annotations are presented to editors.

The Editor role can also be played by a third party system. Systems like Fox [1] and NERD [Rizzo and Troncy, 2012] use ensembles of other systems to decide on which annotation to choose. In such a setting, each individual system being used by Fox or NERD is performing the role of Annotator, while Fox or NERD are performing the role of an automated Editor.

After annotations are created and potentially edited, they will be employed by a consumer to pursue a given objective.

Definition 4. The Consumer is the person or system that will employ the Annotations for a specific Objective.

In practice, the annotation task can be triggered by the Creator, Editor or Consumer alike. However, in any case, we assume that the agent triggering the annotation aims at optimizing the output for its use by the Consumer. Therefore, as a simplification, we will assume that the Consumer – or sometimes just referred to as 'user' – is the person that effectively inputs the content through the system and requests annotations to be performed.

[1] http://fox.aksw.org/

As different objectives may require that the system perform different annotation tasks, Consumers should be empowered to customize the annotation system according to their objectives, through configuration parameters. After receiving the annotations and contrasting with their needs, Consumers may provide feedback and potentially request adjustments to the annotations. An Editor – that may or may not be the same person as the Consumer – can then input the feedback into the System in order to allow for feedback-based improvement on the annotation process.

3.1.2 Objective

When consumers request an automatic annotation task to be performed, they may have very different intentions in mind. These include interlinking pages on a website based on the concepts that they discuss, summarization of important concepts discussed on page, among others. Because needs may vary for each call to the system, it is important to explicitly account for the objective when developing or evaluating a KBT system,

Definition 5. The Objective is a task that can be completed, facilitated or enhanced with the help of Annotations produced by a KBT system. An Objective is said to be achieved if the Consumer employed the resulting Annotations to accomplish some pre-established goal and obtained satisfactory results by the user's own definition of satisfactory.

In practice, the Objective may be simply expressed by a unique identifier that can be used at evaluation time to better understand the system's performance. An explicit objective allows for implementing systems that correlate intent with other features of the model and therefore build policies that perform better for specific intents. It also allows for a fairer cross-task evaluation of systems. On a superficial level, many information extraction tasks can be modeled as a process with textual content as input and a set of entity mentions as output. However, the way the output is interpreted by the consumer may be

very different depending on the objective. Just to cite an example, consider the similarities between the output that may be produced by NER and KE systems. Although there may be a significant overlap between the annotations produced by NER and KE systems, a direct comparison of such systems is bound to be unfair if the objective is not taken into consideration. First, keyphrases include other kinds of concepts besides entities. Second, NER aims at extracting all entities belonging to a set of entity types, irrespective of their importance or keyphraseness. Therefore, an explicit intent in the model allows reuse but clearly describes the reuse limitations.

3.1.3 Textual Content

The Textual Content is a key part of the model as it represents the data input to the system by the Content Creator. It is also one of the core providers of context – the information used by the annotation systems to make annotation decisions.

Text may come from different kinds of media (PDF documents, audio, images, etc.) In the model we propose, when we use Textual Content, we are referring to its representation as a string of characters, as extracted from a given medium, e.g. the transcription of audio files, the characters recognized from an image, or the plain text extracted from an HTML file.

The Textual Content may exhibit particular characteristics depending on the type of discourse, language and original medium. There are different kinds of textual documents: scientific literature, news, encyclopaedic, microposts, speech-to-text, among others. Different types of discourse may require different behavior from annotation systems. Therefore, one of the important features is the type of document. For fairness, this feature must always be explicit in system evaluations, and the ability to generalize beyond one type of document is an important characteristic of adaptability.

Within a KBT system, the text is typically modeled as a vector of words. Transforming a contiguous string of characters into a vector is done through a process of *Tokenization*.

The process of tokenization involves segmentation (breaking the input into words, punctuation, spaces, etc.), and often involves 'cleaning' steps that extract normalizing features from each token – such as a stem, lemma, part-of-speech, etc. The resulting structure modeling the Textual Content contains one dimension for each word type and a weight quantifying the relationship between each word type and the document – e.g. more important words receive larger weights. Additionally, vectors of features for each word in the document are created to store morphological features (e.g. capitalization, prefixes and suffixes), syntactic features (e.g. part-of-speech), etc. Therefore, let us define:

Definition 6. The Textual Content (W) represents a sequence of words extracted from a document or other text container and is modeled as a set of document feature-value pairs $d(W)$ and token feature-value pairs $t(W)$. Examples of document features includes the language of the document and the document type. Examples of token features include the token type, offset, part-of-speech and token count.

3.1.4 Knowledge Base

The Knowledge Base is the backbone of the KBT task. It is a representation of knowledge about a domain (or domains) that delimits the universe of entities and concepts being considered in a KBT task. We will refer to both entities and concepts under the term 'knowledge base resource' or just 'resource', for short. Each resource in a KB is identified by a globally unique identifier. In a KB, resources are described (e.g. OWL Full vs OWL Lite), and to a level of completeness deemed satisfactory, or viable, by the KB authors. The formalism used to represent the KB and its inference capabilities are only tangential to the KBT task. When discussing a KB in relation to a KBT task, it is convenient to see it as the provider of four main components: 1. the 'controlled vocabulary' for tagging, i.e. globally unique identifiers; 2. a type system that organizes identifiers into classes of

'things'; 3. name variations, including synonyms and common misspellings for identifiers; 4. a semantic model, i.e. descriptions of entities that can be used as context to influence the KBT algorithms.

In its simplest form, the KB can be seen simply as a collection of resources, each represented by its own URI – a resource's unambiguous 'name'. Each resource also has one or many names that are used 'on the surface', in text documents on the Web. Conversely, a 'surface form' may be ambiguous, with several candidate interpretations – e.g. 'Washington' can refer to a city, a state or a person. Therefore, we introduce the following definitions:

Definition 7. Resources. Let R be the set of all resources (entities, concepts, things) in a knowledge base.

Definition 8. Surface Forms. Let S be the set of all surface forms (phrases known to have been used as names) for resource $r_i \in R$. Example surface forms include '*city of berlin*' and '*Capital of East Germany*' (among others) for `dbpedia:Berlin`.

Definition 9. Candidates. Let C be a function that maps from a surface form to a set of candidate resources that can be meant by that name. Here, C produces the set of candidates that is declared in the KB, and may be incomplete. KBT systems may implement their own strategies ($C' : S \rightarrow R$) to extend the sets of candidates provided by the KBs.

Knowledge bases often go beyond just defining a set of URIs for resources, and provide semantic descriptions. The most common forms of descriptions are in the form of type hierarchies and inter-resource relationships. Type hierarchies aim at grouping sets of resources that have similar features into 'classes' – i.e. all resources that are cities belong to the class City. Classes may also be grouped into sets of similar classes using inheritance, yielding a hierarchy of classes – e.g. both Soccer Player and Volleyball Player are types

of Athletes, while both Compact Cars and Trucks are types of Vehicles. Other types of relationships between resources can be present in KBs, relating resources with one another – e.g. stating that a given athlete was born in a given city.

Definition 10. The Knowledge Base is a data source KB $= \langle R, S, C, T, M \rangle$ that provides a set of resources R, that can be referenced in text by surface forms from S, such that R and S are related through C. Each resource in R can belong to a type defined in a type system T and relate to other resources through relationships in M. Let T be a set of resource types – i.e. classes – such that resources in R can be organized in a taxonomy. Let M by a set of semantic relationships between resources in R.

3.1.5 System

The KBT system is an automated agent performing annotations according to user-specified requirements. The workflow performed by a KBT system is depicted on Figure 3.3. First, the system performs **Phrase Recognition**, the segmentation of Textual Content in order to recognize the presence of surface forms. For each surface form identified, the system then performs **Candidate Mapping**, the selection of candidate interpretations for identified names. Then, among the candidates identified, the system performs **Disambiguation**, the ranking and decision of which candidate interpretations to select. And finally, the system performs a **Linking** step to adapt the annotations to the desired objective. Formally, let us define:

Definition 11. KBT System is an automated agent able to create a set of Annotations associating knowledge base identifiers to the Text Content according to an Objective. The KBT task performed by the system can be formalized as a function that maps the content W into a set of annotations for resources from the KB that maximizes the probability that

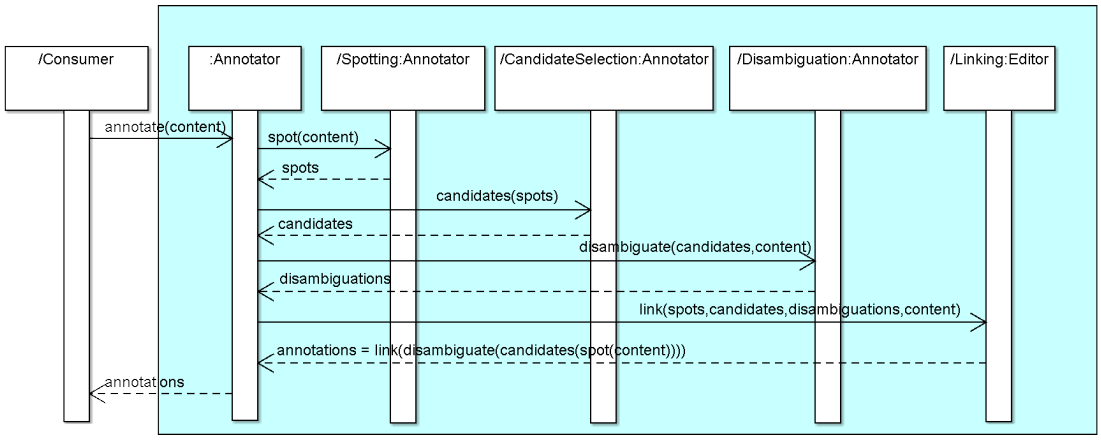


Figure 3.3: Typical internal workflow of a KBT system.

each r found in a position i is useful for an objective O. Let $m_i = \{s_i, r_i, t_i\}$ denote an annotation appearing in position i, having the surface form s_i, and being linked to resource r_i, of type t_i. Note that t_i can be inferred from r_i, and is included in the tuple only for clarity. Then,

$$\begin{aligned} KBT &: W \times O \to \mathcal{P}(R \times S \times \mathbb{N}) \\ KBT(W, O) &= \{m_1, .., m_k \mid m_i = \underset{\{s_i, r_i, t_i\}}{\arg\max}\, P(s_i, r_i, t_i, W, O)\} \end{aligned} \tag{3.1}$$

Thus, KBT is defined as a function mapping the Textual Content W (given an Objective O) to a set of annotations m_i, such that the choice of m_i maximizes the probability of having m_i annotated in W given O.

The KBT task can be further subdivided as illustrated in Figure 3.3, including the subtasks of phrase recognition, candidate mapping, disambiguation and tagging.

Definition 12. Phrase recognition is the process of segmenting the input text into a set of spots S – i.e. a set of phrases such that the likelihood of those phrases being annotated for

a given objective is maximized. Formally:

$$\begin{aligned} \textit{RECOGNITION} &: W \times O \rightarrow \mathcal{P}(S \times \mathbb{N}) \\ \textit{RECOGNITION}(W, O) &= \{s_1, ..., s_n | \arg\max_{s_i} P(s_i, r_i, t_i, W, O)\} \end{aligned} \tag{3.2}$$

Recognition can be seen as a composition of two tasks: segmentation and classification. The most basic recognition tasks will parse the input into segments that are names of entities/concepts in the knowledge base, and segments that are not considered names. Recognition tasks often also include a classification task, which will also attempt to categorize segments into one of the types – with varying granularity – available in the Knowledge Base.

A rudimentary approach for phrase recognition is the enumeration of all n-grams in the input text. This guarantees that, with probability 1, all necessary surface forms will be recognized (among a large number of unnecessary ones) – since all possible surface form mentions will have been enumerated. However, it is a prohibitive approach in its complexity for all but the shortest strings – e.g. for offline batch processing of tweets [Meij et al., 2012]. More sophisticated approaches have the objective of producing a more manageable set of candidate n-grams, while still retaining high recall – i.e. high probability of annotating the surface forms required by the objective. When the objective requires recognizing all entities of pre-determined types – e.g. all people mentioned in the textual content – this task is equivalent to NER, and it includes an extra step of classifying the spots into types. When the objective requires recognizing only the important phrases, this task is equivalent to Keyphrase Extraction (KE). Different objectives may, however, require entities, and general concepts of different levels of importance.

Definition 13. Candidate Mapping is the process of enumerating KB identifiers that are candidate interpretations of a given phrase. It can be formalized as a function that maps a

spot s within the Textual Content W into a set of resources $R_i = \{r_1, ..., r_n\}$ such that the probability that $r \in R_i$ is annotated given objective O is above a threshold δ.

$$\begin{aligned} \textit{CANDIDATES} &: s \times W \times O \to \mathcal{P}(R) \\ \textit{CANDIDATES(s,W,O)} &= \{R_i \,|P(s, r_i, t_i, W, O) > \delta\} \end{aligned} \tag{3.3}$$

Note that the most elementary candidate mapping is the set of all resources in R. Without prior knowledge, a given phrase can mean any of the 'things' in our universe. However, since we assume that KBs always provide at least one surface form for each resource, the most practical approach for candidate mapping relies on retrieving the candidates provided by the KB. Other approaches may, however, expand this mapping with approximations or inferences to account for misspellings, acronyms, abbreviations, etc. as described in Chapter 5. Systems may or may not use the context W when mapping spots to candidate resources. A more informed method can proactively prune (or reweigh) the candidates according to the the context, easing the subsequent disambiguation step.

Definition 14. Disambiguation is the process of choosing the correct interpretation for each spot. Disambiguation can be formalized as a function that maps a spot s appearing in the context W into its corresponding resource $r \in R$ or to NIL if the spot does not refer to a resource in the KB.

$$\begin{aligned} \textit{DISAMBIGUATION} &: r \times W \times O \to R \cup \text{NIL} \\ \textit{DISAMBIGUATION}(s, W, O) &= \{r_i \mid \arg\max_{r_i} P(s_i, r_i, t_i, W, O)\} \end{aligned} \tag{3.4}$$

Disambiguation techniques often include a candidate mapping step in order to reduce the space of possible disambiguations. A second step measures relatedness, as it is assumed that the most closely related candidate to the textual content is the correct disambiguation.

Definition 15. Relatedness is a measure of semantic closeness. The contextual relatedness measures how closely is the semantic definition of an entity related to a given textual content. The entity relatedness measures how close are the semantic definitions of two entities.

$$\textit{CONTEXTUAL RELATEDNESS} : R \times W \rightarrow \mathbb{R} \tag{3.5}$$

$$\textit{ENTITY RELATEDNESS} : R \times R \rightarrow \mathbb{R} \tag{3.6}$$

The disambiguation may be approached locally – where each spot is evaluated separately based on their relatedness with the textual content – or globally, where each spot affects the collective disambiguation decisions based on relatedness between candidate disambiguations. In either case, one corresponding resource is returned for each spot (or NIL in case no corresponding resource exists).

Definition 16. Tagging/Linking is the process of qualifying a relationship between a document and each resource r_i that is pertinent to objective O given the textual content W. Let Rel be the set of relationships that may exist between a textual content and a resource. Thus,

$$\begin{gathered} \textit{TAGGING} : R \times W \times O \rightarrow \mathcal{P}(Rel) \\ \textit{TAGGING}(r, W, O) : \{rel(r, W, o) | \arg\max_{rel} P(s_i, r_i, t_i, W, O)\} \end{gathered} \tag{3.7}$$

The tagging process can be seen as adding to the text references to resources based on relationships such as: (a) **mentions**: "the resource r is explicitly mentioned in the contents of W". (b) **relevant**: "the resource r is relevant for (understanding/using) the contents of text W". (c) **central/aboutness**: "the resource is central to the topic being discussed in the document', or the document is about the resource' (d) **related**: "the resource r is related to

the topic of the text W".

The relation 'mentions' is one with a lower degree of subjectivity. One can ask evaluators to judge if a given surface form s appearing at position i means r. Even in this case, there are very fine distinctions and disagreements between judges can happen (since, you usually do not have access to the mind of the Creator). For example, consider the sentence: "Europeans know how to enjoy life." There are at least two equally plausible senses for the phrase "Europeans." 1. European in the political sense meaning Citizenship of the European Union. 2. European in the ethnical sense meaning Ethnic groups in Europe. In order to alleviate such subjectivity, inter-annotator agreement is usually reported in evaluation experiments.

The relation 'relevant' is more subjective. This subjectivity can be alleviated by explicitly modeling the objective of the annotation task. In the context of IR, Maron [1977] [Maron, 1977] approached relevance by relating it to a probability of satisfaction, measuring if a document D is about a term set T if user X employs T to search for D. Similarly, objective-specific definitions of relevance must be provided for evaluating this relation.

3.1.6 Annotations

The Annotation is the result of the application of a KBT System over the Textual Content in order to connect it to the Knowledge Base.

Definition 17. The Annotation is a tuple associating a Creator, the input Content, the System and the KB. Annotation(Creator, Editor, Content, System, KB, Objective)

The annotations are the output of the KBT task, and will be the objects on which evaluations will be performed. Note that the creator and editor are can be considered metadata attributes of the content, and are omitted from the equations 3.1-3.7 for readability.

Definition 18. An annotation judgement or moderation $J(u, a, r) : \{0, 1\}$ records that a

user u judges that annotation a holds a relationship r between the annotation components $\langle \mathsf{User}, \mathsf{Textual\ Content}, \mathsf{System}, \mathsf{KB}, \mathsf{Objective} \rangle$.

The annotation tasks can now be defined according to these relations. Entity linking can be evaluated as judgements of the relation Mentions(Content,KB). Topic indexing can be evaluated as judgements of the relation Relevant(Content,KB,Objective=Domain). Keyphrase extraction can be evaluated as judgements of the relation Relevant(Content,KB,Objective).

3.2 Conclusion

The conceptual framework presented in this chapter organizes and facilitates the identification of applicable features that influence the quality of semantic annotation with regard to the specific needs of each case. This framework, thus, has two intended consequences: to ease the comparison between existing approaches, and to enable the creation of systems that adapt to the needs of each application. In Section 7 we describe its evaluation in a variety of use cases.

The DBpedia Knowledge Base

The framework proposed in this dissertation is generally applicable to any knowledge base. In order to provide a concrete discussion of the role of a knowledge base in KBT, we will focus on its application in the context of DBpedia. DBpedia [Auer et al., 2007, Bizer et al., 2009, Lehmann et al., 2013] is one of the largest and well known knowledge bases shared as Linked Data on the Web. It has been extensively used in research and applications, and provides cross-domain structured information about millions of entities and concepts. In this chapter the knowledge base is described, along with extensions that were made to DBpedia to support KBT tasks.

4.1 Extracting a Knowledge Graph from Wikipedia

Wikipedia has grown into one of the central knowledge sources of mankind and is maintained by thousands of contributors. Wikipedia articles consist mostly of natural language text, but also contain different types of structured information, such as infobox templates, categorization information, images, geo-coordinates, and links to external Web pages. The DBpedia project extracts various kinds of structured information from Wikipedia editions in multiple languages through an open source extraction framework[1]. It combines all this information into a multilingual cross-domain knowledge base by extracting triples and mapping them to an ontology.

[1] https://github.com/dbpedia/extraction-framework

4.1.1 Extracting Triples

For every page in Wikipedia, a Uniform Resource Identifier (URI) is created in DBpedia to identify an entity or concept being described by the corresponding Wikipedia page. For instance, consider the city of Rio de Janeiro in Brazil. The Wikipedia URL for Rio's page is http://en.wikipedia.org/wiki/Rio_de_Janeiro. DBpedia refers to the city of Rio by the URI http://dbpedia.org/resource/Rio_de_Janeiro (or dbpedia:Rio_de_Janeiro for short), and keeps a link back to the Wikipedia page that describes that resource. During the extraction process, structured information from the wiki such as infobox fields, categories and page links are extracted as RDF triples and are added to the knowledge base as properties of the corresponding URI. Figure Figure 4.1 shows a snippet illustrating triples describing Rio.

```
1 dbpedia:Rio_de_Janeiro dbpo:areaTotal "1.26e+09"^^xsd:schema  .
2 dbpedia:Rio_de_Janeiro dbpo:leaderName dbpedia:Eduardo_Paes .
3 dbpedia:Rio_de_Janeiro dbpo:PopulatedPlace/areaTotal ...
     "1260.0"^^dbpd:squareKilometre .
```

Figure 4.1: A snippet of the triples describing dbpedia:Rio_de_Janeiro.

In order to homogenize the description of information in the knowledge base, a community effort has been initiated to develop an ontology schema and mappings from Wikipedia infobox properties to this ontology. This significantly increases the quality of the raw Wikipedia infobox data by typing resources, merging name variations and assigning specific datatypes to the values. As of March 2012, there were mapping communities for 23 languages[2]. The English Language Wikipedia, as well as the Greek, Polish, Portuguese and Spanish language editions have mapped (to the DBpedia Ontology) templates covering approximately 80% of template occurrences[3]. Other languages such as Catalan, Slovenian, German, Georgian and Hungarian have covered nearly 60% of template occurrences. As

[2]See: http://mappings.dbpedia.org
[3]See: http://mappings.dbpedia.org/index.php/Mapping_Statistics

a consequence, most of the facts displayed in Wikipedia pages via templates are being extracted and mapped to a unified schema.

4.1.2 The DBpedia Ontology

The DBpedia Ontology organizes the knowledge on Wikipedia in 320 classes which form a subsumption hierarchy and are described by 1,650 different properties. It features labels and abstracts for 3.64 million things in up to 97 different languages of which 1.83 million are classified in a consistent ontology, including 416,000 persons, 526,000 places, 106,000 music albums, 60,000 films, 17,500 video games, 169,000 organizations, 183,000 species and 5,400 diseases. Additionally, there are 6,300,000 links to external web pages, 2,724,000 links to images, 740,000 Wikipedia categories and 690,000 geographic coordinates for places.

The alignment between Wikipedia infoboxes and the ontology is done via community-provided mappings that help to normalize name variation in properties and classes. Heterogeneities in the Wikipedia infobox system, like using different infoboxes for the same type of entity (class) or using different property names for the same property, can be alleviated in this way. For example, '*date of birth*' and '*birth date*' are both mapped to the same property `birthDate`, and infoboxes '*Infobox Person*' and '*Infobox FoundingPerson*' have been mapped by the DBpedia community to the class `Person`. DBpedia Mappings currently exist for 23 languages, which means that other infobox properties such as '*data de nascimento*' or '*Geburtstag*' – date of birth in Portuguese and German, respectively – also get mapped to the global identifier `birthDate`. That means, in turn, that information from all these language versions of DBpedia can be merged. Knowledge bases for smaller languages can therefore be augmented with knowledge from larger sources such as the English edition. Conversely, the larger DBpedia editions can benefit from more specialized knowledge from localized editions [Tacchini et al., 2009, Mendes et al., 2012c].

We also extended the coverage of known instance types by importing extra `rdf:type`

statements between DBpedia and the DBpedia Ontology from Aprosio et al., 2013 [Palmero Aprosio et al., 2013], between DBpedia and Freebase[4] and between DBpedia and OpenCyc[5] by Pohl, 2012 [Pohl, 2012].

4.1.3 Cross-Language Data Fusion

As part of the extraction process, DBpedia infers resource equivalence (owl:sameAs) relationships from inter-language links from Wikipedia. For example, if a page on the English-language Wikipedia links to a page on the Portuguese-language Wikipedia as its corresponding page, then there is some evidence that the subjects discussed in these pages are equivalent. If there is also a link from the Portuguese-language to the English-language Wikipedia, then the DBpedia Extraction Framework asserts an equivalence between the corresponding DBpedia resources[6]. This allows applications consuming data from DBpedia to use information from multiple languages to complement one another. Although hopefully complementary, these descriptions originating from Wikipedias edited by different communities may also be conflicting. Some properties that only admit one true value at a time (e.g. the population of a city or its total area) may contain distinct values in each of the different Wikipedia editions. In the context of data integration, the "process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation" is known as Data Fusion [Bleiholder and Naumann, 2009].

In order to demonstrate the challenges and opportunities in fusing data from multiple DBpedias, we conducted an evaluation considering an application interested in processing data from municipalities in Brazil [Mendes et al., 2012c]. According to the Brazilian Institute for Geography and Statistics, there are 5,565 municipalities in Brazil[7]. However, some of this information may be absent from DBpedia due to incomplete or missing Wikipedia

[4] http://downloads.dbpedia.org/3.8/links/freebase_links.nt.bz2

[5] http://opencyc.org

[6] Bijective inter-language links were present in DBpedia until the release 3.8. As of the 3.9 release of DBpedia, inter-language links are obtained from a centralized store provided by WikiData.

[7] http://www.ibge.gov.br/cidadesat

infoboxes, missing mappings in the DBpedia Mapping Wiki or irregularities in the data format that were not resolved by the DBpedia Extraction Framework. Furthermore, a particular Wikipedia edition may be out of date or incorrect. Data integration is commonly applied in order to increase data quality along at least three dimensions: completeness, conciseness and consistency [Bleiholder and Naumann, 2009]. In the following we demonstrate the impact of fusing data from the Portuguese and English DBpedia editions on the completeness, conciseness and consistency of properties `dbpedia:areaTotal`, `dbpedia:foundingDate` and `dbpedia:populationTotal`. For formal definitions of the metrics and more details on the experiment, please refer to [Bleiholder and Naumann, 2009] and [Mendes et al., 2012c].

Completeness On the schema level, a data set is complete if it contains all of the attributes needed for a given task. On the data (instance) level, a data set is complete if it contains all of the necessary objects for a given task. The $Completeness(p)$ for both English and Portuguese-language DBpedia editions are shown on Table 4.1 for each property in our example. The percentages shown represent the completeness of DBpedia English before integration (en), DBpedia Portuguese before integration (pt), and completeness after integration (final). As expected, the integration process increased completeness for both data sets, with particularly high increase (more than 9x) for DBpedia English in the properties `areaTotal` and `populationTotal`. The property `foundingDate` was actually more complete in DBpedia English, and provided an increase of roughly 4% in completeness for DBpedia Portuguese.

Conciseness On the schema level, a data set is concise if it does not contain redundant attributes (two equivalent attributes with different names). On the data (instance) level, a data set is concise if it does not contain redundant objects (two equivalent objects with different identifiers). The intensional conciseness (schema-level) in our example is 1, since both datasets used the same schema. The extensional conciseness (instance-level)

was also 1, since both DBpedia editions only contained one URI per object. Similarly to the case of completeness, we have defined a finer grained *conciseness* metric for a given property p to measure the proportion of objects that do not contain more than one **identical** value for p (redundant), with regard to the universe of unique property values. See Mendes et al. [2012c] for formal definitions. The increase in $Conciseness(p)$ for each of the properties p in our use case is shown on Table 4.1. The properties `areaTotal` and `populationTotal` were 89.80% and 89.51% concise, respectively, while `foundingDate` was 99.66% concise. The integration process yielded an increase in conciseness of roughly 10% for `areaTotal` and `populationTotal`, with only minor increase for `foundingDate` which was already very concise in the original datasets.

Consistency A data set is consistent if it is free of conflicting information. In the context of this paper, the **consistency** of a data set is measured by considering properties with cardinality 1 that contain more than one (distinct) value. The increase in $Consistency(p)$ for each of the properties p in our use case is shown on Table 4.1. The property `foundingDate` had no conflicts observed in the original data. However, we observed an increase in consistency of roughly 10% for `areaTotal` and `populationTotal` which were 90.48% and 90.69% consistent in the original data.

Table 4.1: Impact of the data integration process in quality indicators.

Property p	**Completeness**(p)			**Conciseness**(p)	**Consistency**(p)
	en	pt	final	gain	gain
`areaTotal`	7.31%	71.28%	71.32%	+10.20%	+9.52%
`foundingDate`	4.22%	1.06%	5.27%	+0.34%	-
`populationTotal`	7.58%	71.32%	71.41%	+10.49%	+9.31%

The availability of more complete and cleaner information will benefit a number of applications that rely on these data. Moreover, the transfer of information between DBpedia editions of different sizes and maturity levels will benefit smaller editions by allowing them to import information from the larger and more mature editions.

4.1.4 RDF Links to other Data Sets

DBpedia provides over 27 million RDF links pointing at records in other data sets, and has over 39 million incoming links [Lehmann et al., 2013]. For instance, links to Word Net Synsets [Miller, 1995] were generated by relating Wikipedia infobox templates and Word Net synsets and adding a corresponding link to each entity that uses a specific template. DBpedia also includes links to other ontologies and knowledge bases, including Cyc [Lenat, 1995], Umbel.org, Schema.org and Freebase.com. The high interconnectivity of DBpedia makes it one of the most important datasets on the Linking Open Data Cloud (Figure 4.2).

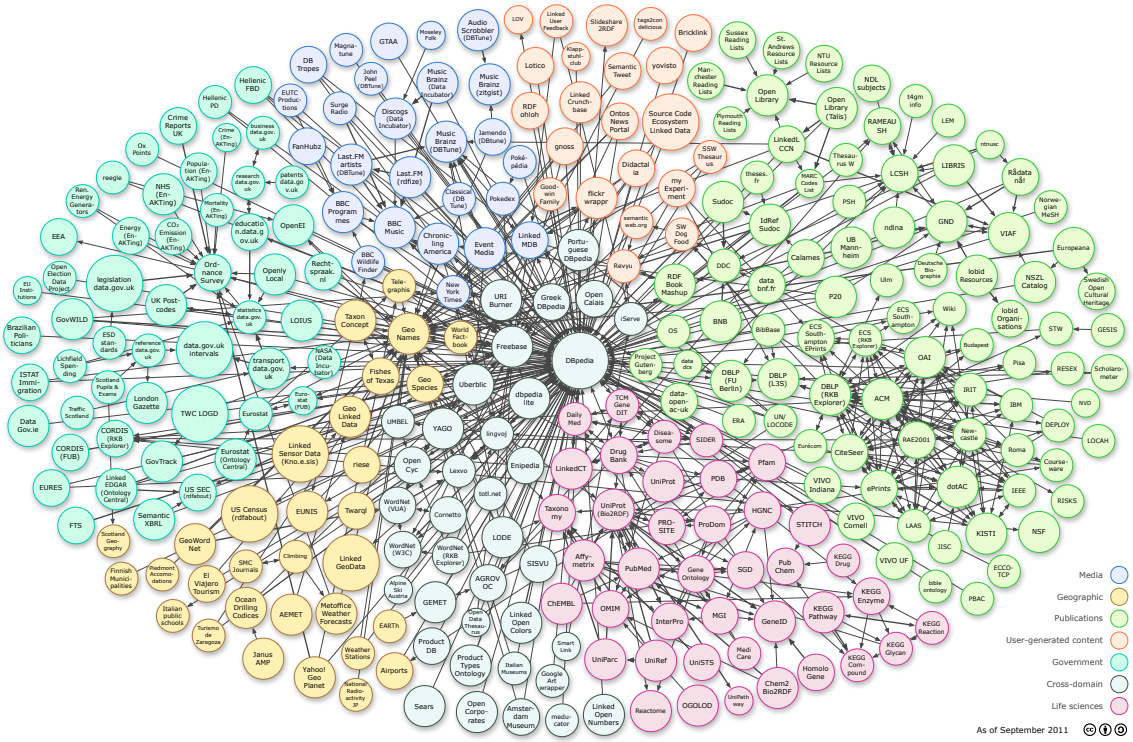


Figure 4.2: Linking Open Data Cloud diagram [Jentzsch et al., 2011].

Some of these resources are particularly useful for supporting information extraction tasks. For instance, Project Gutenberg[8] offers thousands of free e-books that can be used to build corpora. Another example is the New York Times, which began to publish its in-

[8]See: http://www.gutenberg.org/

ventory of articles collected over the past 150 years. As of January 2010, 10,000 subject headings had been shared. The links from DBpedia to authors and texts in Project Gutenberg could be used for backing author identification methods, for instance. Meanwhile, the links to concepts in the New York Times database, enable its usage as an evaluation corpus [Sandhaus, 2008] for Named Entity Recognition and Disambiguation algorithms, amongst others.

4.2 Supporting Natural Language Processing

DBpedia is very useful for KBT and other NLP tasks as it provides a large set of reference global identifiers in a large number of domains. However, DBpedia has historically focused on information that can be extracted from structured parts of Wikipedia. In this section I describe the creation of extended data sets that specifically aim at supporting computational linguistics tasks [Mendes et al., 2012a]. These include the Lexicalization, Topic Signatures, Topical Concepts and Grammatical Gender data sets.

4.2.1 The Lexicalization Data Set

The DBpedia Lexicalization Data Set provides access to alternative names for entities and concepts, associated with several scores estimating the association strength between name and URI. Currently, it contains 6.6 million scores for alternative names.

Three DBpedia data sets are used as sources of name variation: Titles, Redirects and Disambiguation Links[9]. *Labels* of the DBpedia resources are created from Wikipedia page titles, which can be seen as community-approved surface forms. *Redirects* to URIs indicate synonyms or alternative surface forms, including common misspellings and acronyms. As redirects may point to other redirects, we compute the transitive closure of a graph built from redirects. Their labels also become surface forms. *Disambiguation Links* provide

[9] http://wiki.dbpedia.org/Downloads37

ambiguous surface forms that are "confusable" with all resources they link to. Their labels become surface forms for all target resources in the disambiguation page. Note that we erase trailing parentheses from the labels when constructing surface forms. For example the label '*Copyright (band)*' produces the surface form '*Copyright*'. This means that labels of resources and of redirects can also introduce ambiguous surface forms, additionally to the labels coming from titles of disambiguation pages. The collection of surface forms created as a result of this step constitutes an initial set of name variations for the target resources.

We augment the name variations extracted from titles, redirects and disambiguations by collecting the *anchor texts* of page links on Wikipedia. Anchor texts are the visible, clickable text of wiki page links that are specified after a pipe symbol in the MediaWiki syntax (e.g. `[[Apple_Inc.|Apple]]`). By collecting all occurrences of page links, we can create statistics of co-occurrence for entities and their name variations. We perform this task by counting how many times a certain surface form sf has been used to link to a page uri. We calculate the conditional probabilities $p(uri|sf)$ and $p(sf|uri)$ using maximum likelihood estimates (MLE). The pointwise mutual information $pmi(sf, uri)$ is also given as a measure of association strength. Finally, as a measure of the prominence of a DBpedia resource within Wikipedia, $p(uri)$ is estimated by the normalized count of incoming page links of a uri in Wikipedia.

This data set can be used to estimate ambiguity of phrases, to help select unambiguous identifiers for ambiguous phrases, or to provide alternative names for entities, just to cite a few examples. By analyzing the DBpedia Lexicalization Data Set, one can note that approximately 4.4 million surface forms are unambiguous and 392,000 are ambiguous. The overall average ambiguity per surface form is 1.22 – *i.e.* the average number of possible disambiguations per surface form. Considering only the ambiguous surface forms, the average ambiguity per surface form is 2.52. Each DBpedia resource has an average of 2.32 alternative names. These statistics were obtained from Wikipedia dumps using a

```
SELECT ?resource
WHERE {
 ?resource dcterms:subject
      <http://dbpedia.org/resource/Category:Biology> .
}
```

Figure 4.3: SPARQL query demonstrating how to retrieve entities and concepts under a certain category.

script[10] written in Pig Latin [Olston et al., 2008] which allows its execution in a distributed environment using Hadoop[11].

4.2.2 The Thematic Concepts Data Set

Wikipedia relies on a category system to capture the idea of a 'theme', a subject that is discussed in its articles. Many of the categories in Wikipedia are linked to an article that describes the main topic of that category. We rely on this information to mark DBpedia entities and concepts that are 'thematic', that is, they are the center of discussion for a category.

A simple SPARQL query[12] can retrieve all DBpedia resources within a given Wikipedia category (Figure 4.3). A variation of this query can use the Thematic Concepts Data Set to retrieve other DBpedia resources related to certain themes (Figure 4.4). The two queries can be combined with trivial use of SPARQL UNION. This set of resources can be used, for instance, for creating a corpus from Wikipedia to be used as training data for topic classifiers.

[10] Script available at https://github.com/dicode-project/pignlproc

[11] http://hadoop.apache.org

[12] Please note that the wikiPageLinks data set, that is used in Figures 4.4 and 4.4, is not loaded in the public SPARQL endpoint, but is available for download and local usage.

```
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX dbpedia:<http://dbpedia.org/resource/>
SELECT ?resource
WHERE {
?resource dbpedia-owl:wikiPageWikiLink dbpedia:Biology .
}
```

Figure 4.4: SPARQL query demonstrating how to retrieve pages linking to topical concepts.

4.2.3 The Grammatical Gender Data Set

DBpedia contains 416,000 instances of the class Person. We have created a DBpedia Extractor that uses a simple heuristic to decide on a grammatical gender for each person extracted. While parsing an article in the English Wikipedia, if there is a mapping from an infobox in this article to the class dbpedia-owl:Person, we record the frequency of gender-specific pronouns in their declined forms (Subject, Object, Possessive Adjective, Possessive Pronoun and Reflexive) – i.e. he, him, his, himself (masculine) and she, her, hers, herself (feminine).

```
dbpedia:Aristotle       foaf:gender "male"@en .
dbpedia:Abraham_Lincoln foaf:gender "male"@en .
dbpedia:Ayn_Rand        foaf:gender "female"@en .
dbpedia:Andre_Agassi    foaf:gender "male"@en .
dbpedia:Anna_Kournikova foaf:gender "female"@en .
dbpedia:Agatha_Christie foaf:gender "female"@en .
```

Figure 4.5: A snippet of the Grammatical Gender Data Set.

We assert the grammatical gender for each resource being extracted if the number of occurrences of masculine pronouns is higher than the occurrence of feminine pronouns by a margin, and vice-versa. In order to increase the confidence in the extracted grammatical gender, the released data set includes only genders for articles where the frequency of pronouns in the feminine gender is 2 times higher than the masculine (or vice-versa). Furthermore, we experimented with a minimum occurrence of gender-specific pronouns on

one page of 5, 4 and 3. The resulting data covers 68%, 75% and 81%, respectively, of the known instances of persons in DBpedia. Our extraction process assigned the grammatical gender "male" to roughly 85% and "female" to roughly 15% of the people. Figure 4.5 shows example data.

4.2.4 Occurrence Statistics Data Set

The Occurrence Statistics Data Set enables the description of DBpedia Resources in a more unstructured fashion, as compared to the structured factual data provided by the Mapping-based properties. It is based on occurrences of DBpedia entities (mentions) in natural language text. We consider each paragraph on Wikipedia as contextual information to model the semantics of entities under the Distributional Hypothesis [Harris, 1954]. The intuition behind this hypothesis is that entities or concepts that occur in similar contexts tend to have similar meanings.

To create this data set, paragraphs that contain wiki links to the corresponding Wikipedia page of each DBpedia entity or concept are extracted. Paragraphs are tokenized and aggregated, such that statistics can be generated on how many times a specific token has been seen in the same paragraph as each entity. The co-occurrences between tokens and entities can be used subsequently to build a number of models based on the Distributional Hypothesis, relying on different scoring techniques as it will be described in Chapter 5.

4.2.5 The Topic Signatures Data Set

The statistics generated in the Occurrence Statistics Data Set can be employed in a Vector Space Model [Salton et al., 1975] of terms weighted by their co-occurrence with the target entity. In our VSM, each entity is represented by a vector, and each term is a dimension of this vector. Term scores are computed using the TF*IDF weight. We use those weights to select the strongest related terms for each entity and build topic signatures [Lin and Hovy,

```
dbpedia:Alkane sptl:topicSignature "carbon alkanes atoms"
dbpedia:Astronaut sptl:topicSignature  "space nasa"
dbpedia:Apollo_8 sptl:topicSignature  "first moon week"
dbpedia:Actinopterygii sptl:topicSignature  "fish species genus"
dbpedia:Anthophyta sptl:topicSignature "forests temperate plants"
```

Figure 4.6: A snippet of the Topic Signatures Data Set.

2000]. Figure 4.6 shows examples of topic signatures in our data set.

Topic signatures can be useful in tasks such as Query Expansion and Document Summarization [Nastase, 2008]. An earlier version of this data set has been successfully employed to classify ambiguously described images as good depictions of DBpedia entities [García-Silva et al., 2011].

4.3 Conclusion

One of the most important components in a KBT task is the knowledge base, as it forms the universe from which to draw annotations (or to which new discovered phrases can be added). DBpedia is one of the most important KBs on the Web of Data. It provides a wealth of cross-domain multilingual information that is useful for a number of applications, including for connecting textual documents to the Web of Data. By exploiting its roots in Wikipedia, a number of expansions to DBpedia were made to enable the construction of KBT systems. Chapter 5 discusses DBpedia Spotlight, a KBT system that uses data from DBpedia to enable adaptive annotation as a service for a variety of use cases.

DBpedia Spotlight: A System for Adaptive Knowledge Base Tagging of DBpedia Entities and Concepts

In this chapter we present DBpedia Spotlight [Mendes et al., 2011] a system that enables tagging of text corpora with knowledge bases. DBpedia Spotlight is a more comprehensive and adaptable solution than its predecessors. First, while other semantic annotation systems are often restricted to a small number of entity types, such as people, organizations and places, DBpedia Spotlight is able to annotate entities of any of the 360 classes in the DBpedia Ontology. Second, since a single optimized solution is unlikely to fit all task-specific requirements, our system enables user-provided configurations for different use cases with different needs. Users can flexibly specify the domain of interest, as well as the desired coverage and error tolerance for each of their specific annotation tasks.

DBpedia Spotlight takes advantage of the DBpedia ontology for specifying annotation policies. Annotations can be restricted to instances of specific classes (or sets of classes) including subclasses. Alternatively, arbitrary SPARQL queries over the DBpedia knowledge base can be provided in order to determine the set of instances that should be annotated. For instance, consider use cases where users have prior knowledge of some aspects of the text (e.g. dates), and have specific needs for the annotations (e.g. only Politicians). A SPARQL

query can be sent to DBpedia Spotlight in order to constrain the annotated resources to only politicians in office between 1995 and 2000, for instance. In general, users can create restrictions using any part of the DBpedia knowledge base.

Moreover, DBpedia Spotlight considers scores such as prominence (how many times a resource is mentioned in Wikipedia), topical relevance (how close a paragraph is to a DBpedia resource's context) and contextual ambiguity (is there more than one candidate resource with similarly high topical relevance for this surface form in its current context?). Users can configure these parameters according to their task-specific requirements.

Our approach relies on four basic components The *phrase recognition* recognizes in a sentence the phrases that may indicate a mention of a DBpedia resource. *Candidate selection* maps the spotted phrase to resources that are candidate disambiguations for that phrase. The *disambiguation*, in turn, uses the context around the spotted phrase to decide for the most likely choice among the candidates. Finally, during the *tagging* stage, annotation can be customized by users to their specific needs through *configuration* parameters.

5.1 Implementation

The DBpedia Spotlight system is organized in four main modules: core, index, rest, and eval. The core module contains the main interfaces and model objects used to communicate between different parts of the system. It also contains the back-ends, which store the pre-processed data for fast retrieval at annotation time. A conceptual view of the core module is shown in Figure 5.1. The index module contains the code responsible for pre-processing the data and storing them in the back-ends. The rest data contains the interface that interprets HTTP requests, parses the input, and serializes the output in the requested formats. The dist module contains code pertaining to the installation and distribution of the system, while the eval module contains classes and scripts geared towards evaluating the system.

When the rest module receives a request, it triggers a workflow that is typically ex-

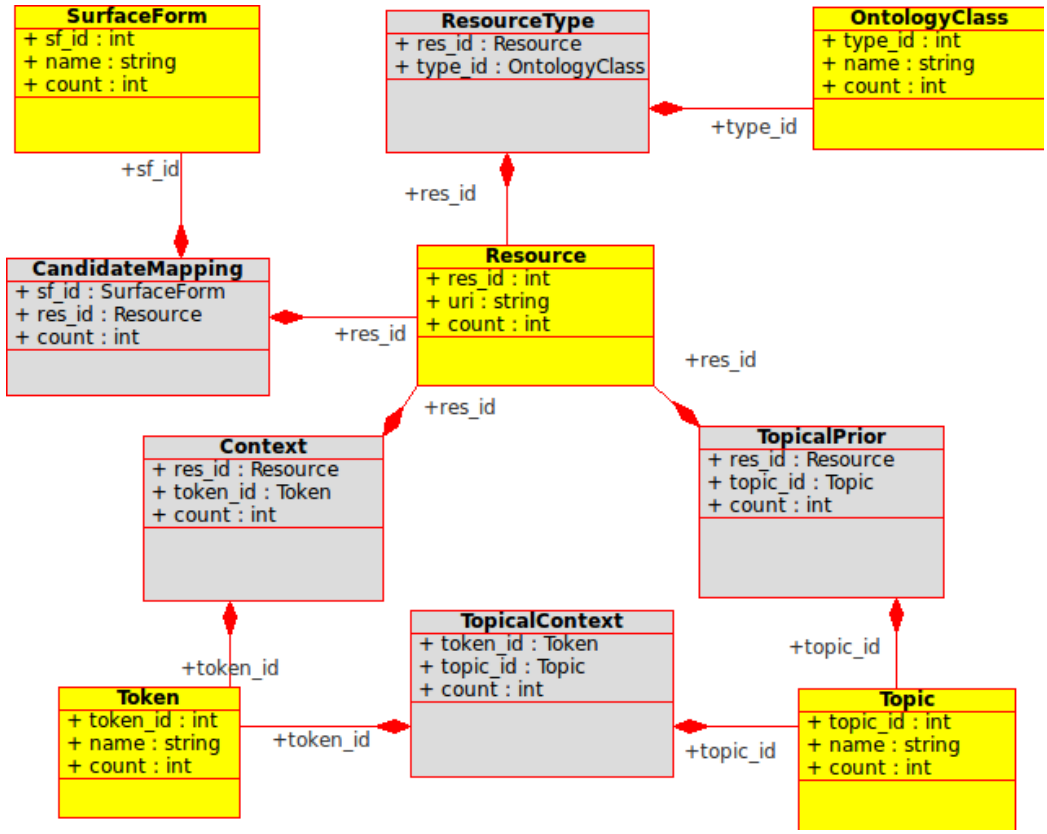


Figure 5.1: Model objects in DBpedia Spotlight's core model.

ecuted in phases: phrase recognition, candidate selection, disambiguation and tagging. However, each phase is self contained and exposed separately as a component in order to provide full flexibility in combining strategies for each phase. In the remainder of this section we explain the implementation of each component.

5.1.1 Phrase Recognition

A naive approach for phrase recognition is the enumeration of all possible token subsequences from length 1 to the number of tokens in the input. This approach is, however, impractical as it generates an exponential number of false positives – i.e. phrases that should not be annotated. Executing the disambiguation step for these phrases would be unnecessary and computationally wasteful.

In this section we describe a number of practical approaches that exploit different characteristics of phrases that commonly constitute annotations in different use cases.

Lexicon-based Recognition (L)

A simple approach for phrase recognition is the usde of a string matching algorithm that relies on a lexicon of name variations for the target terms in the knowledge base. For our Lexicon-based approach, we used the LingPipe Exact Dictionary-Based Chunker [Alias-i, 2011] which relies on the Aho-Corasick string matching algorithm [Aho and Corasick, 1975] with longest case-insensitive match. The lexicon used was obtained from the DBpedia Lexicalization Dataset [Mendes et al., 2011, 2012a].

Because the lexicon-based phrase recognition does not select phrases with regard to their context but merely searches for any phrases known as possible DBpedia entities/concepts, this step still produces a high number of false positives. One example is the set of function words that have entries on Wikipedia, but whose annotation would be undesirable in use cases such as blog annotation, because it would confuse the reader with too many

(arguably unnecessary) links. However, eliminating those phrases from the lexicon upfront is not an option, as they may have other significant meanings – e.g. the word '*up*' can be a function word in some contexts, but it can also refer to `Up(2009 film)`, a movie by Pixar.

Noun-phrase chunk heuristic (L_{NP*})

In many use cases, the objective of annotation is to mark the *things* being talked about in text. Therefore, a simple heuristic to eliminate false positives early in the process is to only annotate terms that are within noun phrases. We therefore extended the Lexicon-based phrase recognizer with a simple heuristic that only allows phrases that contain at least one noun. This annotation style would disregard general concepts such as `Running` and `Crying` when they appear as verbs, but would include concepts like `Perpetual war`. The suitability of this heuristic depends on the use case.

Filtering common words (CW)

In the case of Wikification, only notable entities should be annotated. The Wikipedia guidelines for link creation[1] focus on encouraging the annotation of non-obvious references that help with understanding an encyclopaedic article. The guidelines explicitly instruct users to "avoid linking plain English words".

Therefore, we used a classifier to attempt the detection and removal of common words at phrase recognition time [Daiber, 2011]. We used two main classifiers: a classifier for single-token spot candidates and a multi-token spot candidate classifier. Both classifiers try to find words that carry common knowledge meaning, however the multi-token spot candidate classifier focuses mainly on detecting phrases that appear in syntactically irregular positions.

In a manual classification of 4497 mentions in encyclopedic and newspaper texts,

[1] http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(linking)

50.05% of all mentions were classified as common word occurrences (including annotations of verbs, adjectives, idioms and fixed phrases). Based on this dataset we trained two Bayesian Network models. This choice was empirical – we chose the model that provided the best results in our preliminary tests. Features included neighboring tokens, POS and n-grams. A detailed description of the method and implementation are available online [Daiber, 2011].

Keyphrase Extraction (KE)

In other use cases (e.g. blogs) one would like to identify only important phrases in the context of the document or website. We use Kea [Frank et al., 1999], a supervised algorithm to identify candidate keyphrases in a document collection. It relies on the Naïve Bayes algorithm and features such as TF*IDF and token distance in order to learn its 'keyphraseness' from a training set of known keyphrases. Admittedly, the fairest evaluation for this spotter would have included a cross-validation experiment over the evaluation set. However, we are interested in testing its fitness for an unsupervised, general-purpose online annotation task. Therefore we use Kea with the default prediction model distributed by its authors.

Named Entity Recognition (NER)

In use cases such as the annotation in online newspapers, we commonly see the focus on entities of specific types (e.g. people and organizations), whereas keyphrases such as '*foreign policy*' are less commonly annotated [2]. For use cases that focus on a small set of common term types (such as People, Location, Organization) it is viable to apply named entity recognizers as a strategy for phrase recognition. Therefore we also tested a NER-backed phrase recognizer based on the default models distributed with OpenNLP 1.5.1[3].

[2] http://nyti.ms/qsYAyt

[3] http://opennlp.sourceforge.net/models-1.5/

NER extended by noun phrase n-grams (NER ∪ NP)

We provide also a hybrid approach mixing named entities and more general terms within noun phrase chunks, based on previous work [Ratinov et al., 2011]. We consider as only the expressions marked as named entities by the NER phrase recognizer, the noun-phrase chunks extracted by a NP chunker, and all sub-expressions of up to 5 tokens of the noun-phrase chunks. This increases the coverage of the NE phrase recognizer, which tends to generate fewer annotations. This implementation is also based on standard models distributed with the OpenNLP project.

Custom Recognizers

The system is also able to process phrases that have been recognized externally. Users interested in running their own custom recognizers can encode the detected surface forms in an XML format dubbed SpotXml (Figure 5.2).

```
1 <annotation text="Brazilian oil giant Petrobras and U.S. oilfield ...
    service company Halliburton have signed a technological ...
    cooperation agreement, Petrobras announced Monday. The two ...
    companies agreed on three projects: studies on contamination ...
    of fluids in oil wells, laboratory simulation of well ...
    production, and research on solidification of salt and carbon ...
    dioxide formations, said Petrobras. Twelve other projects are ...
    still under negotiation.">
2     <surfaceForm name="oil" offset="10"/>
3     <surfaceForm name="company" offset="56"/>
4     <surfaceForm name="Halliburton" offset="64"/>
5     <surfaceForm name="oil" offset="237"/>
6     <surfaceForm name="other" offset="383"/>
7 </annotation>
```

Figure 5.2: Example phrases recognized externally and encoded as SpotXml.

5.1.2 Candidate Selection

We follow the spotting with a candidate selection stage in order to map resource names to candidate disambiguations (e.g. *Washington* as reference to a city, to a person or to a state). We use the DBpedia Lexicalization dataset for determining candidate disambiguations for each surface form.

The candidate selection offers a chance to narrow down the space of disambiguation possibilities. Selecting fewer candidates can increase time performance, but it may reduce recall if performed too aggressively. Due to our generality and flexibility requirements, we decided to employ minimal pre-filtering and postpone the selection to a user-configured post-disambiguation configuration stage.

The candidate selection phase can also be viewed as a way to pre-rank the candidates for disambiguation before observing a surface form in the context of a paragraph. Choosing the DBpedia resource with highest prior probability for a surface form is the equivalent of selecting the "default sense" of some phrase according to its usage in Wikipedia. The prior probability scores of the lexicalizations dataset, for example, can be utilized at this point. We report the results for this approach as a baseline in Chapter 6.

5.1.3 Disambiguation

After selecting candidate resources for each surface form, our system uses the context around the surface forms, e.g. paragraphs, as information to find the most likely disambiguations.

We modeled DBpedia resource occurrences in a Vector Space Model (VSM) [Salton et al., 1975] where each DBpedia resource is a point in a multidimensional space of words. In light of the most common use of VSMs in Information Retrieval (IR), our representation of a DBpedia resource is analogous to a document containing the aggregation of all paragraphs mentioning that concept in Wikipedia. Similarly, the TF (Term Frequency) weight

is commonly used in IR to measure the local relevance of a term in a document. In our model, TF represents the relevance of a word for a given resource. In addition, the Inverse Document Frequency (IDF) weight [Jones, 1972] represents the general importance of the word in the collection of DBpedia resources.

Albeit successful for document retrieval, the IDF weight fails to adequately capture the importance of a word for disambiguation. For the sake of illustration, suppose that the term '*U.S.A*' occurs in only 3 concepts in a collection of 1 million concepts. Its IDF will be very high, as its document frequency is very low (3/1,000,000). Now suppose that the three concepts with which it occurs are `dbpedia:Washington_D.C.`, `dbpedia:George_Washington`, and `dbpedia:Washington_(U.S._State)`. As it turns out, despite the high IDF weight, the word '*U.S.A*' would be of little value to disambiguate the surface form '*Washington*', as all three potential disambiguations would be associated with that word. IDF gives an insight into the global importance of a word (given all resources), but fails to capture the importance of a word for a specific set of candidate resources.

In order to weigh words based on their ability to distinguish between candidates for a given surface form, we introduce the Inverse Candidate Frequency (ICF) weight. The intuition behind ICF is that the discriminative power of a word is inversely proportional to the number of DBpedia resources it is associated with. Let R_s be the set of candidate resources for a surface form s. Let $n(w_j)$ be the total number of resources in R_s that are associated with the word w_j. Then we define:

$$ICF(s, w_j) = \log \frac{|R_s|}{n(w_j)} = \log |R_s| - \log n(w_j) \tag{5.1}$$

The theoretical explanation for ICF is analogous to Deng et al. [2009], based on Information Theory. Entropy [Shannon, 1951] has been commonly used to measure uncertainty in probability distributions. It is intuitive that the discriminative ability of a context word

should be inversely proportional to the entropy, i.e. a word commonly co-occurring with many resources is less discriminative overall. With regard to a word's association with DBpedia resources, the entropy of a word can be defined as:

$$H(w) = H(P(R|w)) = -\sum_{i \in R_s} p(r_i|w) \log p(r_i|w)$$

Suppose that the word w is connected to those resources with equal probability

$$p(r|w) = \frac{1}{n(w)} = \frac{1}{|R_s|}$$

Thus, the maximum entropy is transformed to:

$$H'(w) = -\sum_{i \in R_s} \frac{1}{n(w)} log(1/n(w)) \tag{5.2}$$

$$= -\sum_{i \in R_s} \frac{1}{n(w)} (log(1) - log(n(w))) \tag{5.3}$$

$$= -\sum_{i \in R_s} \frac{1}{|R_s|} (-log(n(w))) \tag{5.4}$$

$$\log n(w) \tag{5.5}$$

Since generally the entropy tends to be proportional to the frequency $n(w)$, we use the maximum entropy to approximate the exact entropy in the ICF formula. This simplification has worked well in our case, simplifying the calculations and reducing storage and search time requirements.

Given the VSM representation of DBpedia resources with TF*ICF weights, the disambiguation task can be cast as a ranking problem where the objective is to rank the correct DBpedia resource at position 1. Our approach is to rank candidate resources according to the similarity score between their context vectors and the context surrounding the surface form. As a result, candidate resources that appeared in similar contexts to the input text will

```
[{"Umbro":1.2643485069274902},
 {"Kappa_(company)":1.0901415348052979},
 {"Lotto_Sport_Italia":1.0321451425552368},
 {"Puma_AG":0.8806803226470947},
 {"Le_Coq_Sportif":0.8236550092697144},
 {"Reebok":0.7530333399772644},
 {"Fila_(company)":0.6978422999382019},
 {"Sportswear":0.6948583126068115},
 {"Nike,_Inc.":0.6457382440567017}]
```


Figure 5.3: Results for a request to `/related` for the top-ranked URIs by relatedness to `dbpedia:Adidas`. Each key-value pair represents a URI and the relatedness score (TF*IDF) between that URI and Adidas.

have higher similarity scores and naturally move to the higher ranked positions. We use cosine similarity, a widely used measure in information retrieval, as the similarity measure in our implementation.

5.1.4 Relatedness

Entity relatedness can be approached similarly to the contextual similarity used by the disambiguation implementation. We model each resource r under the distributional semantics hypothesis by extracting a vector (in a VSM) of all words that have occurred around r. The relatedness between two resources r_1 and r_2 is then the cosine similarity between the vectors for r_1 and r_2. Figure 5.3 shows the output of a request to the `/related` endpoint requesting the top URIs ranked according to their relatedness to `dbpedia:Adidas`.

5.1.5 Tagging

Many of the current approaches for annotation tune their parameters to a specific task, leaving little flexibility for users to adapt their solution to other use cases. Our approach is to generate a number of metrics to inform the users and let them decide on the policy

that best fits their needs. In order to decide whether to annotate a given resource, there are several aspects to consider: can this resource be confused easily with another one in the given context? Is this a commonly mentioned resource in general? Was the disambiguation decision made with high confidence? Is the resource of the desired type? Is the resource in a complex relationship within the knowledge base that rules it out for annotation? The offered configuration parameters are described next.

Resource Set to Annotate. As editors – and sometimes consumers – have insight into the types of entities that should be annotated, it is important to provide a parameter that allows them to focus the output onto a subset of the KB. In our case the available types are derived from the class hierarchy provided by the DBpedia Ontology. Users can provide whitelists (allowed) or blacklists (forbidden) of URIs for annotation. Whitelisting a class will allow the annotation of all direct instances of that class, as well as all instances of subclasses. Support for SPARQL queries allows even more flexibility by enabling the specification of arbitrary graph patterns. There is no restriction to the complexity of relationships that a resource must fulfil in this configuration step. For instance, the user could choose to only annotate concepts that are related to a specific geographic area, time period in history, or are closely connected within the Wikipedia category system.

Resource Prominence. For many applications, the annotation of rare or exotic resources is not desirable. For example, the `Saxon_genitive` (*'s*) is very commonly found in English texts to indicate possession (e.g. Austria*'s* mountains are beautiful), but it can be argued that for many use cases its annotation is rather uninformative. An indicator for that is that it has only seven Wikipedia inlinks. With the *support parameter*, users can specify the minimum number of inlinks a DBpedia resource has to have in order to be annotated.

Topic Pertinence. The topical relevance of the annotated resource for the given context can be measured by the similarity score returned by the disambiguation step. The score is higher for paragraphs that match more closely the recorded observations for a DBpedia resource. In order to constrain annotations to topically related resources, a higher threshold

for the topic pertinence can be set.

Contextual Ambiguity. If more than one candidate resource has high topical pertinence to a paragraph, it may be harder to disambiguate between those resources because they remain partly ambiguous in that context. The difference in the topical relevance of two candidate resources to a paragraph gives us an insight on how "confused" the disambiguation step was in choosing between these resources. The score is computed by the relative difference in topic score between the first and the second ranked resource. Consumers dealing with applications that require high precision may decide to reduce risks by not annotating resources when the contextual ambiguity is high.

Disambiguation Confidence. We define a confidence parameter, ranging from 0 to 1, of the annotation performed by DBpedia Spotlight. This parameter takes into account factors such as the topical pertinence and the contextual ambiguity. Setting a high confidence threshold instructs DBpedia Spotlight to avoid incorrect annotations as much as possible at the risk of losing some correct ones. We estimated this parameter on a development set of 100,000 Wikipedia samples. The rationale is that a confidence value of 0.7 will eliminate 70% of incorrectly disambiguated test cases. For example, given a confidence of 0.7, we get the topical pertinence threshold that 70% of the wrong test samples are below. We integrate that with the contextual ambiguity score by requiring a low ambiguity when the confidence is high. A confidence of 0.7, therefore, will only annotate resources if the contextual ambiguity is less than $(1 - confidence) = 0.3$.

5.2 Using DBpedia Spotlight

DBpedia Spotlight is available both as a Web Service and via a Web Application. While the Web Application allows users to get acquainted with the system's functions on a Web browser, the Web Service allows third-party applications to programmatically reuse the system.

5.2.1 Web Application

By using the Web application, users can test and visualize the results of the different service functions. The interface allows users to configure confidence, support, and to select the classes of interest from the DBpedia ontology. Text can be entered in a text box and, at user's request, DBpedia Spotlight will highlight the surface forms and create associations with their corresponding DBpedia resources. Figure 5.4 shows an example of a news article snippet after being annotated by the system.

Figure 5.4: DBpedia Spotlight Web Application.

5.2.2 Web Service

In order to facilitate the integration of DBpedia Spotlight into external web processes, we implemented REST and SOAP web services for the annotation and disambiguation processes. The snippet below shows an example call to the REST API.

Service endpoints: The web service interface allows access to Phrase Recognition, Candidate Selection and Disambiguation/Tagging operations. The endpoint **/spot** takes text as

```
curl http://spotlight.dbpedia.org/rest/annotate \
  --data-urlencode "text=Pop star Michael Jackson has..." \
  --data "confidence=0.2" \
  --data "support=20"
```

Figure 5.5: Example call to the Web Service using cURL.

input and performs phrase recognition according to an implementation chosen through the parameter **&spotter**. The **/disambiguate** endpoint takes input that has been pre-processed by a custom phrase recognizer (or manually recognized phrases) and encoded as Wiki-Markup or SpotXml (Figure 5.2). The **/annotate** endpoint performs phrase recognition, candidate selection, disambiguation and tagging steps. It may also take input that has been pre-processed externally by a custom spotter. The input in this case can be encoded as Wiki Markup or SpotXml. The **/candidates** endpoint peforms phrase recognition, candidate selection and disambiguation steps. Although it computes the top-ranked candidate (the disambiguation), it also returns a ranked list of all resources and their contextual relatedness scores so that clients can reuse that information as needed. The **/related** endpoint takes as input a URI and returns the top URIs ranked by their relatedness to the input URI.

Input parameters: The web service also allows access to the configuration parameters of our approach. Thus, in addition to confidence, support and DBpedia classes, it accepts SPARQL queries for the DBpedia knowledge base to select the set of resources that are going to be used when annotating.

Output formats: DBpedia Spotlight is able to return HTML+RDFa, XML, JSON or RDF (in NIF) output where each DBpedia resource identified in the text is related to the text chunk where it was found. This flexibility is implemented through content negotiation. Users interested in a particular serialization of the output only need to send an "Accept" header in their HTTP request informing the mime type of the desired output format, and the server complies with the requested serialization. The XML fragment presented below

shows part of the annotation of the news snippet shown in Figure 5.4.

```
<Annotation text="Pop star Michael Jackson..."
   confidence="0.3" support="30"
   types="Person,Place,...">
   <Resources>
     <Resource URI="dbpedia:Michael_Jackson"
       support="5761"
       types="MusicalArtist,Artist,Person"
       surfaceForm="Michael Jackson" offset="9"
       similarityScore="0.31504717469215393" />
     ...
   </Resources>
</Annotation>
```

Figure 5.6: Example XML fragment resulting from the annotation service.

5.2.3 Installation

DBpedia Spotlight is Open Source software and was written mostly in Scala – with some parts in Java, and auxiliary scripts in Bash and Python. The codebase is available from GitHub[4]. Therefore, as an alternative to using the system as a service from its currently deployed endpoint, one may install and use the system within the boundaries of their private networks. This addresses eventual concerns with having sensitive data sent over the network to an external Web service. When installed, the system will have the same capabilities and can used in the same way as the publicly deployed web service.

The build process is managed through Maven, which allows the description of library dependencies and automates the acquisition, installation and deployment of the required software. The Debian packaging system – one of the most used packaging and deployment systems – is also used as an alternative option for installation, facilitating the installation in Linux-based systems. The Debian packages in DBpedia Spotlight are shared as part of the LOD2 Stack [Auer et al., 2012].

[4]DBpedia Spotlight on GitHub: http://github.com/dbpedia-spotlight/

5.3 Continuous Evolution

Knowledge Bases are in constant evolution, as new entities are added, facts about them change, and so on. As the knowledge evolves, so should a KBT system. In this section, we discuss three aspects of evolution of KBT systems. First, KBT systems should be updated as the available knowledge from the KB changes. Second, KBT systems should be able to learn from user feedback, so that automatic annotations that are manually corrected by users will teach the system to reduce future mistakes. Third, and finally, how can DBpedia Spotlight be employed in a virtuous cycle of semantic enhancement that closes a loop with its learning source in order to generate better context for learning in the future.

5.3.1 Live updates

When a new president gets elected, a celebrity gets married, or a world cup is won, Wikipedians rush to edit the world encyclopedia to reflect this new knowledge. DBpediaLive [Hellmann et al., 2009][Morsey et al., 2012] is a project that runs the DBpedia Extraction Framework (DEF) on the Wikipedia stream of modifications and keeps DBpedia in sync with its knowledge source. These modifications also have an impact on DBpedia Spotlight. Besides the creation of new entities in the target knowledge base, new contextual clues may be added as users create new paragraphs or modify existing ones. The modifications on redirects and disambiguation pages also provide name variations, and should be used by DBpedia Spotlight when trying to link phrases to their unique identifiers in the knowledge base.

Two design decisions ease the task of updating DBpedia Spotlight as its source data is modified:

- the use of SPARQL to allow for expressing tagging requirements allows querying the DBpediaLive endpoint, gaining access to the freshest extraction at all times;
- the use of an annotation model that allows incremental updates to scores for candidate

mapping and contextual scoring. Every new page added generates a new entry to the KB, while every new link added to Wikipedia generates an update to the statistics stored for the target entity (both for lexicalizations and for the distributional semantic model).

5.3.2 Feedback incorporation

Users that rely on DBpedia Spotlight for automatically suggesting annotations, can use several GUIs for adjusting wrong annotations, or including missing ones. These user modifications can be sent back to the system in order to help DBpedia Spotlight to retrain exactly on the points where it is making mistakes or missing annotations. There are several scenarios where there are clear incentives for correcting annotations. For example, consider the scenario where a user is annotating its blog post in order to enrich it with related tags. The system suggests annotations, and the user has the chance to accept or correct the suggested annotation. At every user action, feedback is posted back to the DBpedia Spotlight API informing the system if the user accepted, deleted or modified a given tag. In this scenario, users do not consciously think about improving DBpedia Spotlight. Instead, they are improving the organization and linking of their information resources, with a side-effect of improving DBpedia Spotlight for future annotation jobs.

5.3.3 Round-trip semantics

Our vision is that KBT systems should foster a virtuous cycle of semantic enhancement [Héder and Mendes, 2012]. In this cycle, an editor interface is supported by the knowledge base to create better documents, which in turn increases the quality and size of the knowledge base as a whole. With a better and larger knowledge base, KBT systems have more data from which to learn, and can therefore provide better assistance to the user, creating a positive feedback loop (Figure 5.7).

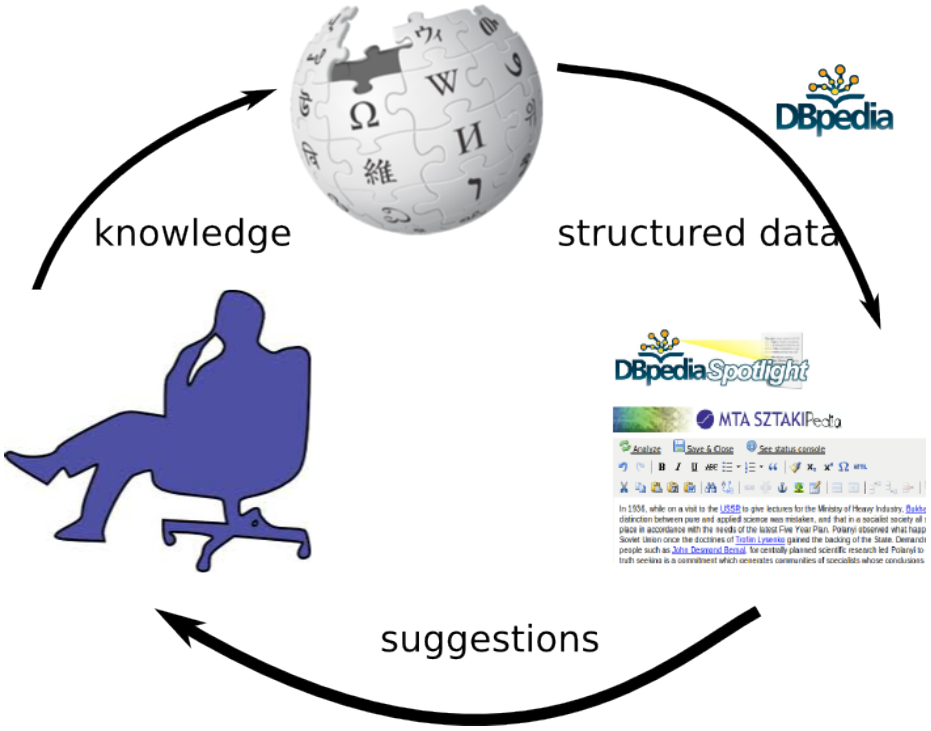


Figure 5.7: Round-trip Semantics

The KBT system's role is to make suggestions and recommendations to improve the knowledge source's amount and quality of information. This could be done by suggesting new links between pages that mention one another, or suggesting correction of links that were added to the wrong target pages. The system may be proactive – triggering a suggestion whenever an improvement opportunity is detected – or act upon user request. An important element of this vision is that the user should make most of the decisions about the suggestions. As a consequence the system must be present online in the editor interface of the user, e.g. as a plugin.

We use the Sztakipedia toolbar[5] to support this knowledge-enhancement cycle on Wikipedia [Héder and Mendes, 2012]. In the first step, structured data which is extracted from Wikipedia is used to construct automatic KBT engines. Those engines can be used to interconnect knowledge in structured and unstructured information sources on the Web, including Wikipedia itself. Sztakipedia-toolbar is a MediaWiki user script which brings

[5]http://pedia.sztaki.hu

DBpedia Spotlight and other kinds of machine intelligence into the Wiki editor interface to provide enhancement suggestions to the user. The suggestions offered by the tool focus on complementing knowledge and increasing the availability of structured data on Wikipedia. This will, in turn, increase the available information for the content enhancement engines themselves, completing a virtuous cycle of knowledge enhancement.

5.4 Conclusion

This chapter presented DBpedia Spotlight, a tool to detect mentions of DBpedia resources in text. The annotations provided by DBpedia Spotlight enable the enrichment of websites with background knowledge, impacting applications such as faceted browsing in text documents and enhanced search capabilities, among others. The main advantage of our system is its comprehensiveness and flexibility, allowing one to configure each step of the annotation workflow according to the desired KBT task. The system allows configuration through the DBpedia ontology, as well as through parameters such as the prominence, contextual ambiguity, topical pertinence and confidence scores. The list of desired types to be identified can be expressed by a list of resource types or by more complex relationships within the knowledge base through SPARQL queries.

A project page with news, documentation, downloads, demonstrations and other information is available at http://spotlight.dbpedia.org.

Core Evaluations

6.1 Evaluation Corpora

Evaluation datasets often capture different aspects of annotation tasks. While some focus on specific entity types, other seek exhaustive annotations of all entities of all types. In the following subsections we present the datasets that we used for evaluating the core components of our system.

6.1.1 Wikipedia

Wikipedia provides a wealth of annotated data that can be used to evaluate our system on a large scale. We randomly selected 155,000 wikilink samples and set aside as test data. In order to really capture the ability of our system to distinguish between multiple senses of a surface form, we made sure that all these instances have ambiguous surface forms. We used the remainder of the samples collected from Wikipedia (about 69 million) as DBpedia resource occurrences providing context for disambiguation as described in Chapter 5.

6.1.2 CSAW

The CSAW project [Kulkarni et al., 2009] provides an evaluation corpus in which texts from random webpages were manually annotated with Wikipedia page ids. During the

manual generation of the dataset, "volunteers were told to be as exhaustive as possible and tag all possible segments, even if to mark them as NA." The dataset contains 7652 annotations referred to as *NA*, for which users explicitly indicated that there is an entity or concept, but that no suitable representative page was found in Wikipedia.

6.2 Spotting Evaluation Results

We used the annotations produced in the CSAW project [Kulkarni et al., 2009] in which texts from random webpages were manually annotated with Wikipedia page ids.

We conducted an evaluation to assess the precision and recall of each phrase recognizer. We model each phrase as a tuple (phrase, phrase offset) in order to account for phrase repetition in the document. Let S be the set of spotted phrases generated by a given phrase recognizer. Let A be the set of annotated phrases in the gold standard. The phrase recognition **precision** is $P = |S \cap A|/|S|$, the proportion of recognized phrases that are correct. The phrase recognition **recall** is the proportion of phrases in the gold standard that were found by the spotter: $R = |S \cap A|/|A|$.

The most important evaluation measure in the context of this section is arguably the recall. As subsequent steps rely on input obtained from phrase recognition, the overall recall of an annotation system can only be at maximum equal to the recall of phrase recognition. Better precision in phrase recognition is also a desirable feature, as a large amount of false positives may degrade time performance, and errors in boundary detection can negatively influence the accuracy of the disambiguation step.

Table 6.1 shows the phrase recognition evaluation results. Note that the precision values of the phrase recognition strategies for this dataset are generally low since the CSAW dataset does not exhaustively annotate every potential entity mention.

The best recall was obtained by the NER $\cup$ NP phrase recognizer. However, it generates a large number of false positives that would be unnecessarily sent for disambiguation.

spotter	P	R	time per spot
$L_{>3}$	4.89	68.20	0.0279
$L_{>10}$	5.05	66.53	0.0246
$L_{>75}$	5.06	58.00	0.0286
L_{NP*}	5.52	57.04	0.0331
$NP_{L*>3}$	6.12	45.40	1.1807
$NP_{L*>10}$	6.19	44.48	1.1408
$NP_{L*>75}$	6.17	38.65	1.2969
CW	6.15	42.53	0.2516
Kea	1.90	61.53	0.0505
NER	4.57	7.03	2.9239
NER $\cup$ NP	1.99	68.30	3.1701

Table 6.1: Evaluation results.

The CW phrase recognizer was able to reduce the number of generated phrases in roughly half (from ≈168K to ≈83K), but at the cost of ≈26% loss in recall. The L_{NP*} phrase recognizer had a smaller loss in recall (≈11%), but only removed ≈25% of the false positives. The keyphrase extractor had the third best recall, but it generated a higher percentage of false positives as compared to most approaches.

For the lexicon-based approaches, we tested lexica of different sizes. For the lexicon $L_{>3}$, we included all name variations for entities or concepts that are the target of at least 3 links on Wikipedia. Similarly, we applied thresholds 10 and 75, yielding $L_{>10}$ and $L_{>75}$ respectively. We observed that the occurrences of page links in Wikipedia follow a power-law distribution – few entities have many links and many entities have very few links. Reducing the lexicon according to the number of links allows drastic savings in main-memory storage, while focusing on the most common concepts.

6.2.1 Error Analysis

In this section we discuss some of the most common mistakes made by the phrase recognizer implementations tested.

Stopwords. While completely ignoring stopwords in phrase recognizers is not feasible due to relevant entities that can be confused with stopwords, a better strategy for detecting stopword occurrences is needed. Namely, case sensitive treatment and using POS tags are within our plans. While the common word detection (L − CW) managed to detect many of these stopwords, it also removed other common words that were deemed "annotation worthy" by the CSAW dataset, such as: soul, eating, specific, used and neat.

Phrase boundaries. In phrases such as "the Internet", "the government" and "the embryo" the CSAW dataset includes the determiner as part of the phrase. In our lexicon, the determiners are not included in the lexicon. Roughly 2% of all phrases in CSAW start with determiners.

Notability. Since Wikipedia enforces notability guidelines, phrase recognizer approaches such as NER are bound to make mistakes by correctly identifying people that are not on Wikipedia – e.g. Tawana Lebale, who was a participant in Big Brother Africa but does not have a Wikipedia page dedicated to her. For these cases, a combination of NER and keyphraseness is planned.

6.3 Disambiguation Evaluation Results

In this evaluation, we were interested in the performance of the disambiguation stage. A spotted surface form, taken from the anchor text of a wikilink, is given to the disambiguation function[1] along with the paragraph that it was mentioned in. The task of the disam-

[1] in our implementation, for convenience, the candidate selection can be called from the disambiguation

Disambiguation Approach	*Accuracy*
Baseline Random	17.77%
Baseline Default Sense	55.12%
Baseline TF*IDF	55.91%
DBpedia Spotlight TF*ICF	73.39%
DBpedia Spotlight Mixed	80.52%

Table 6.2: Accuracies for each of the approaches tested in the disambiguation evaluation.

biguation service is to select candidate resources for this surface form and decide between them based on the context.

The results for the baselines and DBpedia Spotlight are presented in Table 6.2. The performance of the baseline that makes random disambiguation choices confirms the high ambiguity in our dataset (less than 1/4 of the disambiguations were correct at random). Using the prior probability to choose the default sense performs reasonably well, being accurate in 55.12% of the disambiguations. This is indication that our evaluation set was composed by a good balance of common DBpedia resources and less prominent ones. The use of context for disambiguation through the default scoring of TF*IDF obtained 55.91%, while the TF*ICF score introduced in this work improved the results to 73.39%.

The performance of TF*ICF is an encouraging indication that a simple ranking-based disambiguation algorithm can be successful if enough contextual evidence is provided.

We also attempted a simple combination of the prior (default sense) and TF*ICF scores, which we called DBpedia Spotlight Mixed. The mixing weights were estimated through a small experiment using linear regression over held out training data. The results reported in this work used mixed scores computed through the formula:

$$
\begin{aligned}
Mixed(r, s, C) = & \\
& 1234.3989 * P(r|s) \\
& +0.9968 * contextualScore(r, s, C) \\
& -0.0275
\end{aligned} \tag{6.1}
$$

The prior probability $P(r|s)$ was calculated as described in Section 4.2.1. The contextual score used was the cosine similarity of term vectors weighted by TF*ICF as described in Section 5.1.3. Further research is needed to carefully examine the contribution of each component to the final score.

6.3.1 TAC-KBP

In this section we present the evaluation of DBpedia Spotlight on the TAC KBP Entity Linking Task [2]. The objective of this task is to find the correct identifier from a knowledge base (*KB*) of persons, locations and organizations, given a surface form (e.g. an entity name) and the text in which this name occurred. In case there is no such identifier (the entity is not in the *KB*), the system should return NIL. Moreover, the system is required to cluster entities referring to the same NIL entity.

Data Preparation. The TAC KBP knowledge base uses custom entity identifiers (e.g. `E0456437`). The definition for each entity in the knowledge base contains the URL segment of the corresponding article in the English Wikipedia (e.g `A._K._Antony`). As DBpedia also derives its URIs from the Wikipedia pages, we used the page titles to map the TAC KBP knowledge base to DBpedia. However, the versions of Wikipedia used by the TAC KBP knowledge base and DBpedia are different. This means that some articles

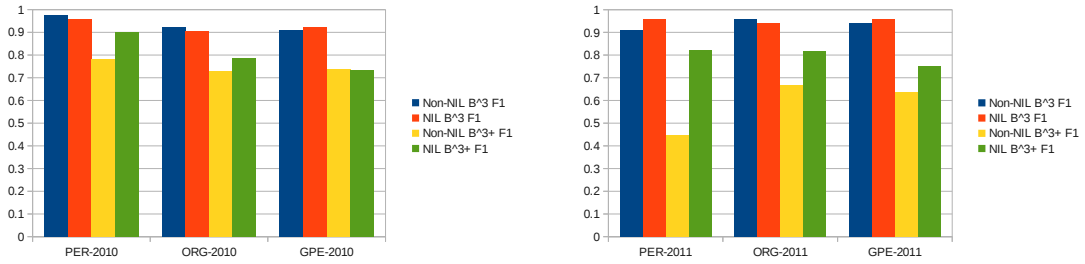
[2] http://nlp.cs.qc.cuny.edu/kbp/2011/

may have become redirect pages, requiring a pre-processing step to map these two KBs. We perform this mapping using the graph of redirects from DBpedia 3.6 (based on the Wikipedia dump from 11/10/2010).

Linking. As DBpedia contains more entities than the TAC KBP knowledge base (*KB*), it is possible that the highest ranking entity after the disambiguation stage is not present in the *KB*. If that is the case, or if no disambiguation was found, this entity is considered NIL. NIL clustering is performed as follows: i) if the entity is in DBpedia, but not in TAC KBP, we use the DBpedia URI to place all references in to the same NIL cluster, and ii) if the entity is not on DBpedia, we use the surface form to perform the clustering.

Data Set	μAVG	B^3P	B^3R	B^3F_1	$B^{3+}P$	$B^{3+}R$	$B^{3+}F_1$
TAC-KBP 2010 EL-1.0	0.827	0.904	0.958	0.930	0.773	0.805	0.789
TAC-KBP 2011 EL-1.1	0.727	0.920	0.971	0.945	0.693	0.713	0.703

Table 6.3: Evaluation results for TAC KBP English Entity Linking Gold Standards.



(a) 2010 (b) 2011

Figure 6.1: Comparison of B^3 and B^{3+} F_1 scores for NIL and Non-NIL annotations, for each entity type for 2010 and 2011 data sets.

Results. The best results achieved by DBpedia Spotlight are presented in Table 6.3. The scores reported are the KBP2010 micro-average (μAVG) [McNamee et al., 2010], the B-cubed cluster scoring (B^3) [Bagga and Baldwin, 1998] and the B-Cubed+ modification

(B^{3+}) [Ji et al., 2011]. We report all three scoring metrics in order to allow comparisons with systems from other years.

Figure 6.1 compares the F1 by entity type for NIL and Non-NIL annotations. A NIL annotation means that an entity was marked in text, but it is not present in the *KB*. The entity types in the *KB* are PER (Person), ORG (Organizations) and GPE (Geopolitical Entity).

6.4 Annotation Evaluation Results

6.4.1 News Articles

We created a manually annotated evaluation dataset from a news corpus to assess completeness of linking, besides accuracy of disambiguation. We created an annotation scenario in which the annotators were asked to add links to DBpedia resources for all phrases that would add information to the provided text.

Our test corpus consisted of 35 paragraphs from New York Times documents from 8 different categories. In order to construct a gold standard, each evaluator first independently annotated the corpus, after which they met and agreed upon the ground truth evaluation choices. The ratio of annotated to not-annotated tokens was 33%.

We compared our results on this test corpus with the performance of publicly available annotation services: OpenCalais[3], Zemanta[4], Ontos Semantic API[5], The Wiki Machine[6], Alchemy API[7] and M&W's wikifier [Milne and Witten, 2008]. Linking to DBpedia is supported in those services in different levels. Alchemy API provides links to DBpedia and Freebase among other sources. Open Calais and Ontos provide some limited linkage between their private identifiers and DBpedia resources. As of the time of writing, Ontos

[3] http://www.opencalais.com
[4] http://www.zemanta.com
[5] http://www.ontos.com
[6] http://thewikimachine.fbk.eu
[7] http://www.alchemyapi.com

only links people and companies to DBpedia. For the cases where the systems were able to extract resources but do not give DBpedia URIs, we used a simple transformation on the extracted resources that constructed DBpedia URIs from labels - e.g. 'apple' becomes `dbpedia:Apple`. We report results with and without this transformation. The results that used the transformation are labeled Ontos+Naïve and Open Calais+Naïve. The service APIs of Zemanta, The Wiki Machine and M&W do not explicitly return DBpedia URIs, but the URIs can be inferred from the Wikipedia links that they return.

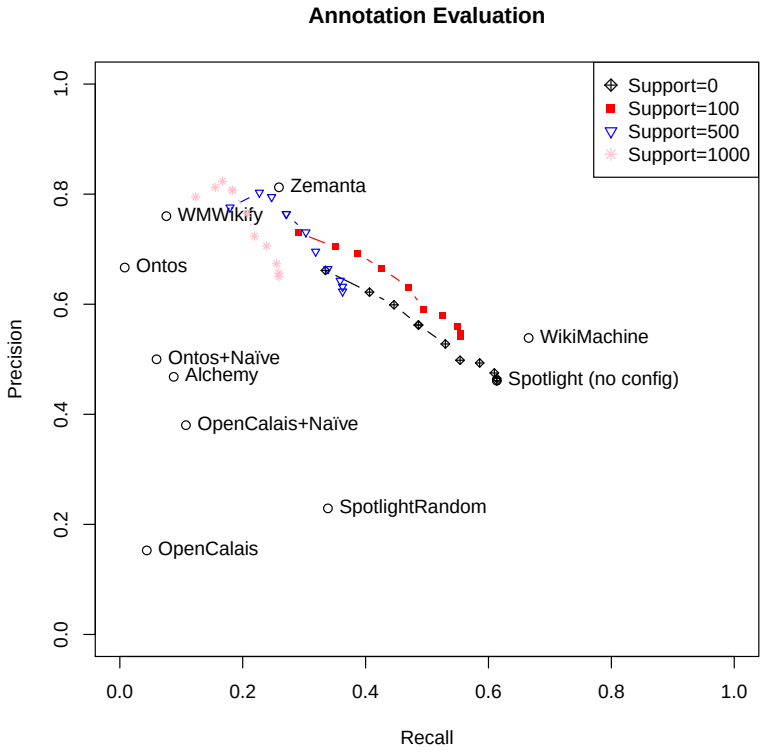


Figure 6.2: DBpedia Spotlight with different configurations (lines) in comparison with other systems (points).

Results

Retrieval as well as classification tasks exhibit an inherent precision-recall trade-off [Buckland and Gey, 1994]. The configuration of DBpedia Spotlight allows users to customize

System	*F1*
DBpedia Spotlight (best configuration)	56.0%
DBpedia Spotlight (no configuration)	45.2%
The Wiki Machine	59.5%
Zemanta	39.1%
Open Calais+Naïve	16.7%
Alchemy	14.7%
Ontos+Naïve	10.6%
Open Calais	6.7%
Ontos	1.5%

Table 6.4: F_1 scores for each of the approaches tested in the annotation evaluation.

the level of annotation to their specific application needs. Figure 6.2 shows the evaluation results. Each point in the plot represents the precision (vertical axis) and recall (horizontal axis) of each evaluation run. The lines show the trade-off between precision and recall as we vary the confidence and support parameters in our service. Each line represents one value of support (varying from 0 to 1000). Each point in the line is a value of confidence (0.1 to 0.9) for the corresponding support. It can be observed that higher confidence values (with higher support) produce higher precision at the cost of some recall and vice versa. This is encouraging indication that our parameters achieve their objectives.

The shape of the displayed graph shows that the performance of DBpedia Spotlight is in a competitive range. Most annotation services lay beneath the F_1-score of our system with every confidence value. Table 6.4 shows the best F_1-scores of each approach. The best F_1-score of DBpedia Spotlight was reached with confidence value of 0.6. The Wiki-Machine has the highest F_1-score, but tends to over-annotate the articles, which results in a high recall, at the cost of low precision. Meanwhile, Zemanta dominates in precision, but has low recall. With different confidence and support parameters, DBpedia Spotlight is able to approximate the results of both WikiMachine and Zemanta, while offering many other configurations with different precision-recall trade-offs in between.

Linking. We estimated the confidence parameter on a development set of 100,000 Wikipedia samples. The rationale is that a confidence value of 0.7 will eliminate 70% of incorrectly disambiguated test cases. For example, given a confidence of 0.7, we get the topical pertinence threshold that 70% of the wrong test samples are below. We integrate that with the contextual ambiguity score by requiring a low contextual ambiguity when the confidence is high. A confidence of 0.7, therefore, will only annotate resources if the contextual ambiguity is less than $(1 - confidence) = 0.3$.

6.5 A Framework for Evaluating Difficulty to Disambiguate

Entity references using ambiguous names are not equally difficult to disambiguate. Consider the case of '*berlin*'. Although there are more than 30 places with that name[8], the vast majority of mentions to 'berlin' on the Web are referring to the entity identified by `dbpedia:Berlin`, the capital of Germany. In this case, we say that `dbpedia:Berlin` is the default sense of '*berlin*'. Intuitively, mentions to such entities should be easier to disambiguate because we can just blindly pick the default sense (without analyzing the context of the mention) and be right in the majority of cases. Meanwhile, disambiguating mentions to other (less popular) senses may require a more meticulous analysis of the context of the mention, and thus may be harder to disambiguate. The validation of this hypothesis with state-of-the-art AaaS systems is one of the central points of this section.

We define a measure of dominance to quantify the 'default sense'-ness of entity mentions through a score ranging from 0 to 1. By definition, the 'default sense' is the entity with the highest dominance score for a name. Meanwhile, an entity with lower dominance is less commonly meant by that name – i.e. it is 'dominated' by other senses.

We use this measure to compare the disambiguation accuracy of well-known and widely used extraction services on the Web, such as Open Calais, Alchemy API, Yahoo!

[8] See the disambiguation page for Berlin on Wikipedia.

Content Analysis Platform, and Zemanta. The results from these tools are further compared with two approaches employed by DBpedia Spotlight.

Definition 19. Confusability. Let the true confusability of a surface form s be the number of meanings that this surface form can have. As new places, organizations and people (just to cite a few examples) are being named every day, and as we do not have access to an exhaustive collection of all named entities in the world, the true confusability of a surface form is unknown. Let the confusability estimate of a surface form be a function $A(s) : S \rightarrow \mathbb{N}$ that maps from a surface form to an estimate of the size of its candidate mapping, so that $A(s) = |C(s)|$.

The confusability of a place name offers only a rough, *a priori* estimate of how difficult it may be to disambiguate that surface form. After observing annotated occurrences of this surface form in a collection, we can make more informed estimates. Some places are expected to be more prominent than others (e.g. larger cities are much more 'talked about' than small towns). Therefore, a strategy that always chooses the most prominent entity as disambiguation is likely to perform reasonably well (by definition). Similarly, given a surface form, some senses are much more dominant than others – e.g. for the name 'berlin', the resource `dbpedia:Berlin` (Germany) is much more 'talked about' than `Berlin,_New_Hampshire` (USA). Thus let us define the estimates Prominence and Dominance as:

Definition 20. Prominence. Let the true prominence of a resource r_i be the amount to which r_i is more well known than other resources $r_k \in R$. Let the prominence estimate $Pr(r_i)$ be the relative frequency that the resource r_i appears linked on Wikipedia with

respect to the frequency of all other resources in R. Formally:

$$Pr(r_i) = \frac{\sum_{s \in S} |WikiLinks(s, r_i)|}{\sum_{s \in S, r \in R} |WikiLinks(s, r)|}$$

Definition 21. Dominance. Let the true dominance of a resource r_i for a given surface form s_i be a measure of how commonly r_i is meant when s_i is used in a sentence. Let the dominance estimate $D(r_i, s_i)$ be the relative frequency with which the resource r_i appears in Wikipedia links where s_i appears as the anchor text. Formally:

$$D(r_i, s_i) = \frac{|WikiLinks(s_i, r_i)|}{\forall_{r \in R} |WikiLinks(s_i, r)|}$$

In an experimental setting, evaluation corpora that contain resources with high ambiguity, low dominance and low prominence are considered more difficult. This is due to the fact that a corpus with these characteristics requires a more careful examination of the context of each mention before algorithms can choose the most likely disambiguation. In cases with low ambiguity, high prominence and high dominance, simple baselines that ignore the context of the mention can be already fairly successful.

6.5.1 Comparison with annotation-as-a-service systems

To provide a concrete assessment of the proposed measures, we compared a number of popular extraction services available online. We have provided the same blog text from our gold standard to each API and tested if the returned annotations contain the place annotated in the gold standard. Formally, for each blog post $b_i = (t_i, r_i)$ in our gold standard, we provide the text t_i to an API and retrieve a set of resources R_{api} extracted by that API. For each blog post, the extraction accuracy is 1 if the URI $r_i \in R_{api}$, and 0 otherwise. The

overall extraction accuracy is measured by the mean of the individual extraction accuracy over all blog posts B, for $N = |B|$:

$$ACCURACY = \frac{\sum_i^N |\{r_i\} \cap R_{api}|}{N}$$

Annotation Service	accuracy	accuracy (hard)
Alchemy API	27.63	23.26
Open Calais	18.32	12.79
DBpedia Spotlight TF*ICF	47.45	**36.05**
DBpedia Spotlight GEM	**59.76**	**37.79**
Yahoo! CAP	26.43	8.72
Zemanta	**59.76**	33.14

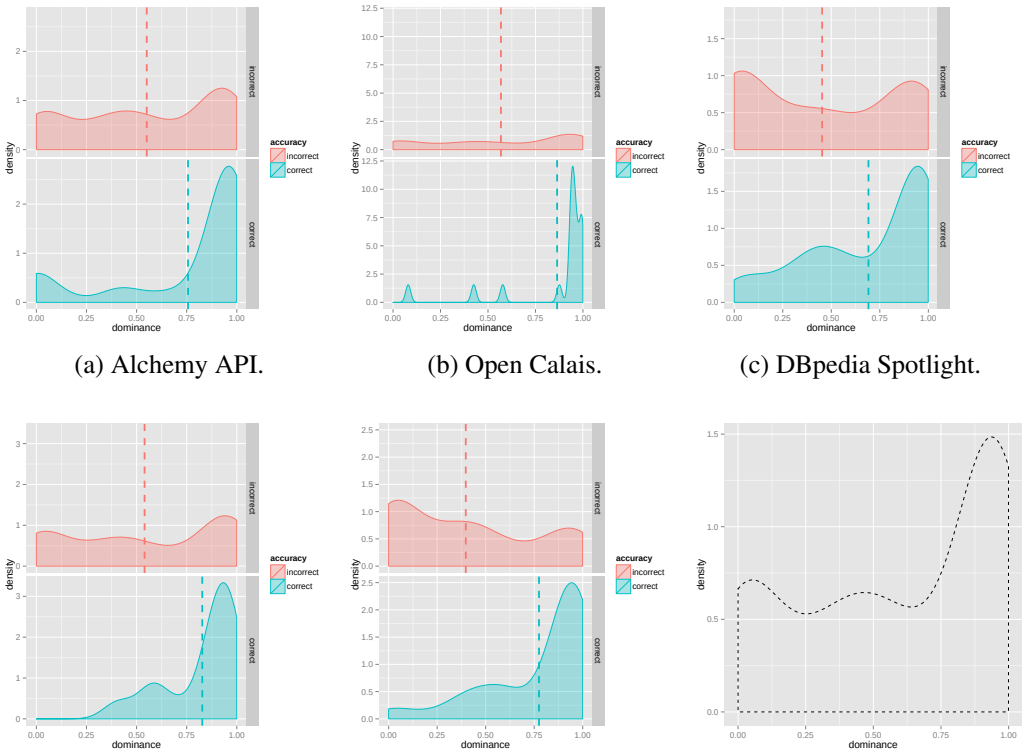
Table 6.5: Extraction accuracy on our gold standard for well known extraction services.

Table 6.5 compares the accuracy of existing services such as Zemanta, Open Calais, Alchemy API and Yahoo!CAP. While Zemanta and DBpedia Spotlight GEM [Daiber et al., 2013] obtain the best accuracy on the entire dataset, Zemanta's accuracy degrades more severely when we focus on the more difficult examples.

We are interested in evaluating if the tested systems perform equally well on difficult disambiguations as compared with those with lower difficulty scores. Figures 6.3a to 6.3e show an analysis of the dominance for correct and incorrect disambiguations made by each of the analyzed systems. For each system, the top box (red area) shows the distribution of dominance scores within the incorrect disambiguations. The bottom box (blue area), shows the dominance distribution for correct disambiguations.

The plots show a general skew to the top right, as expected, due to the underlying distribution of dominance scores – recall that most of the examples in the gold standard have high dominance (Figure 6.3f.)

However, we can notice that there are consistently more correct disambiguations than incorrect ones when dominance is higher. Conversely, there are less correct examples and



(a) Alchemy API. (b) Open Calais. (c) DBpedia Spotlight.

(d) Yahoo CAP. (e) Zemanta. (f) GoldStandard.

Figure 6.3: For each annotation set, the figure shows the distribution of dominance scores for incorrect (red, top box) and correct (green, bottom box) examples. Showing mentions for phrases with $A(s) > 1$.

more incorrect ones when dominance is lower. This is particularly the case for Yahoo! CAP (Figure 6.3d) and Zemanta (Figure 6.3e), which seem to have much worse accuracy with examples below the mean dominance. These findings also apparent from Table 6.5, which shows the accuracy of each system on the toughest disambiguations.

These observations are consistent with our intuition, as it would make sense for large, general-purpose providers such as Yahoo! CAP and Zemanta to optimize for more 'common' subjects. However, we have shown that there is a significant portion of the blogosphere that demands more careful evaluation of the context, before a disambiguation decision can be made. For those cases, more robust techniques or more contextual information are warranted.

6.6 Conclusions

In this chapter we described evaluation results using a variety of corpora used in the semantic annotation community. We discussed evaluations in light of our framework, providing detailed results for each of the steps in the annotation process.

Case Studies

In this chapter we describe case studies that were performed to validate our conceptual model and system. For each use case, we outline the most relevant dimensions of our conceptual framework. We describe how these dimensions can be used in order to adapt the behavior of a system to the needs of each of the use cases described.

7.1 Named Entity Recognition in Tweets

The task proposed through the MSM2013 Concept Extraction dataset is equivalent to the task performed by Named Entity Recognition (NER) systems. Recall that according to our Conceptual Model Chapter 3, this task can be broken down into two problems. First, a segmentation problem requires finding boundaries of entity names within sentences; and second, a classification problem requires correctly classifying the segment into one of the entity types. We have tested approaches that perform each task separately, as well as approaches that perform both tasks jointly.

We were interested in evaluating DBpedia Spotlight's adaptability to NER on microposts. First, we tested an unsupervised approach – i.e. one that does not use the training set provided in the challenge. It uses DBpedia Spotlight's phrase recognition and disambiguation to perform NER in a two-step process of segmentation and classification (dbpedia_spotlight_1.tsv). For this approach, the example microposts were sent through DBpedia Spotlight's lexicon-based recognition, and subsequently through the disambigua-

tion algorithm. Based on the types of the entities extracted, we used our manual mappings to classify the names into one of the NER types.

Second, we tested a joint segmentation/classification method based on a supervised-machine learning approach enhanced by knowledge-based distant supervision from DBpedia. We use lexicalizations from DBpedia to indicate that a given token may be within an entity or concept name. This feature is intended to help with the segmentation task, particularly in cases where morphological characteristics of a word are not informative. Moreover, we use DBpedia Spotlight to suggest DBpedia Resources for the microposts and extract ontology types from those resources to create a battery of features that further biases the classification task towards the correct types.

We collected all our best features and created a Linear-Chain Conditional Random Fields (CRF) model to act as our NER (dbpedia_spotlight_2.tsv). We used Factorie [McCallum et al., 2009] to implement our CRF. Our features include morphological (e.g. punctuation, word shape), context-based (e.g. surrounding tokens) and knowledge-based characteristics. Our knowledge-based features include the presence of a token within a name in our knowledge base, as well as the types predicted for this entity.

Given those features and the provided training corpus, the model is trained using stochastic gradient ascent. Gibbs sampling is used to estimate the posterior distribution for each label during training. We also added a small post-processing filter to remove whole entities that contain less than two letters or digits in them as well as entities with name ”the” and ”of”.

Finally, we included Stanford NER [Finkel et al., 2005] as our third baseline (dbpedia_spotlight_3.ts since it is a well known NER implementation.

Table 7.1 presents our evaluation results on the training set. Precision, recall and F1 on Table 7.1 were computed based on the overlap (using exact name and type matches) between the set of entities we extracted and the set of annotated entities. The scores shown for our supervised method are our averaged 10-fold cross-validation scores.

Syst./NERType	PER	LOC	ORG	MISC	**Average**
•	P / R / F1	P / R / F1	P / R / F1	P / R / F1	P / R / F1
Unsup. (1)	0.95 0.50 0.65	0.62 0.58 0.60	0.62 0.38 0.47	0.22 0.21 0.21	0.60 0.42 0.48
CRF (2)	0.86 0.66 0.75	0.82 0.7 0.76	0.73 0.56 0.63	0.49 0.29 0.36	0.72 0.53 0.61

Table 7.1: Comparison between NER approaches on the MSM2013 Challenge Training Set.

We also report token-based precision, recall and F1 averaged over a 10-fold cross-validation on the training set. For Stanford NER (Vanilla) (with default features), we obtain P: 0.77, R: 0.54 and F1: 0.638. For Stanford NER (Enhanced), after adding our knowledge-based features, we observe improvements to P: 0.806, R: 0.604 and F1: 0.689. The same evaluation approach applied to DBpedia Spotlight CRF yields P:0.91, R:0.72, F1:0.8.

We found the segmentation to be far harder than classification in this dataset. First, as expected in any task that requires agreement between human experts, some annotation decisions are debatable. Second, inconsistent tokenization was a big issue for our implementation.

In some cases, our model found annotations that were not included by the human-annotators, such as `ORG/twitter`, where "twitter account" could be (but was not) interpreted as an account within the `ORG` Twitter. In other cases, our model trusted the tokenization provided in the training set and predicted `MISC/Super Bowl-bound` while the human-generated annotation was `MISC/Super Bowl`.

However, in general, after guessing correctly the boundaries, the type classification seemed an easier task. Our manual mappings already obtain an average accuracy over 82%. After training, those numbers are improved even further.

However, in some cases, there seems to be some controversial issues in the classification task. Is "Mixed Martial Arts" a `Sport` or a `SportEvent`? Is "Hollywood" an organization or a location? Depending on the context, the difference can be subtle and may be missed even by the human annotators.

By far, the toughest case to classify is `MISC`. Perhaps, such a "catch all" category may be too fuzzy, even for human annotators. The annotations often contain human languages like `MISC/English;MISC/Dutch;` where the guidelines stated that only Programming languages would be annotated.

7.2 Managing Information Overload

Every day, Web users are using social media to simultaneously publish millions of microblog posts – microposts – with opinions, observations and suggestions that may represent invaluable information for businesses and researchers around the world[1]. Taking advantage from this "wisdom of the crowd" – which refers to "the process of taking into account the collective opinion of a group of individuals rather than a single expert to answer a question"[2] –, Twitter data has been successfully used, for example, to forecast box-office revenues for movies [Asur and Huberman, 2010] or to manage earthquakes detection [Sakaki et al., 2010].

However, the vast amount of microblog data published each second can be overwhelming for users that are interested in very specific topics and events. Hashtags for user-provided content categorization, and keyword search are common instruments used with the intent to alleviate information overload. However, these solutions have limitations on the kinds of filtering that they enable. It becomes very difficult, for example, to use only keywords and hashtags narrow a stream of information down to only people in a certain region of the globe that are expressing negative sentiments about competitors of a given brand of interest.

In order to perform streaming, aggregation and distribution of the tagged content, we implemented Twarql, an information aggregation and filtering service enabled by KBT

[1] See statistics from Twitter at http://blog.twitter.com/2010/02/measuring-tweets.html.

[2] From Wikipedia, see http://en.wikipedia.org/wiki/Wisdom_of_the_crowd.

to help users select microposts through the use of more expressive queries. Through our approach, users can subscribe to more flexible "concepts" (Figure 7.1) instead of only hashtags or users. We adapt the definition of concept from Tom Mitchell [Mitchell, 1997]. Each such concept can be viewed as describing some subset of objects or events defined over a larger set (e.g. subset of tweets that mention competitors of a product), or alternatively, each concept can be thought of as a boolean-valued function defined over this larger set (e.g., a function defined over all tweets, whose value is true for tweets containing competitors of a product and false for all others). In our work, concepts are defined by SPARQL queries through user-friendly interfaces that do not require any knowledge of query language from the users. The system then publishes the evolving relevant content as an annotated web feed, proactively delivered through a push model, providing a simple interface for users to follow topic-relevant information in real-time (Figure 7.1).

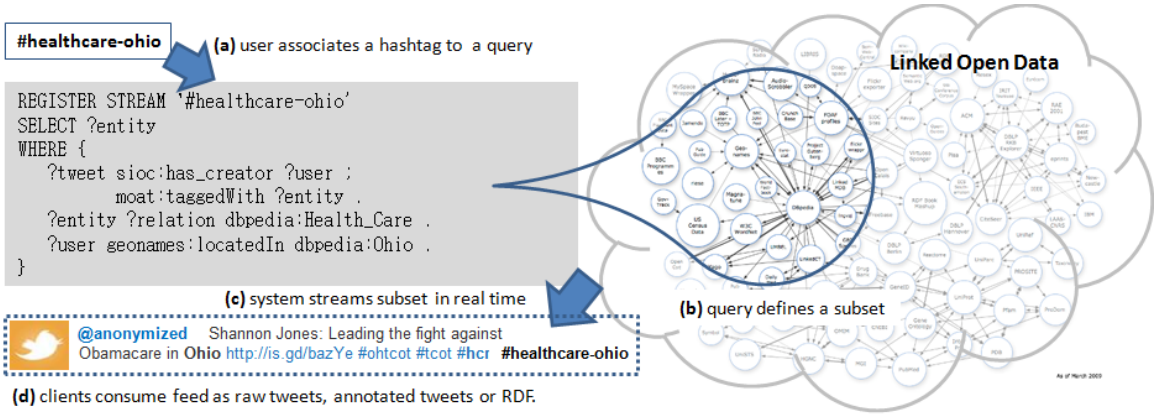


Figure 7.1: Concept feeds are named by a hashtag and define a subset of tweets through a query.

To illustrate how KBT is can be used, consider for example, a brand tracking scenario. Social media has become an ubiquitous platform for Internet users to quickly and openly voice their opinions. The Nielsen Global Online Consumer Survey (July 2009) reports that 90% of people trust recommendation from their social network and 70% trust recommendations posted online [The Nielsen Company, 2009]. Product managers, marketers and investors interested in monitoring the health and the value of their brands have social me-

dia as a rich source to engage in brand tracking: monitoring and analyzing the reputation of a brand. A few questions that one my ask in this scenario include: 1. Where do people like my product the most? 2. What do people think about competitors of this product? Users may encode these questions as SPARQL queries (Sections 7.2 to 7.2) and instruct the system to perform KBT under those constraints.

We list below a few example queries to illustrate how KBT supports Twarql in a brand tracking scenario[3].

Example 1: Location. "Give me a stream of locations where my product is being mentioned right now."

```
1 SELECT ?location
2 WHERE {
3   ?tweet moat:taggedWith dbpedia:IPad .
4   ?presence opo:currentLocation ?location .
5   ?presence opo:customMessage ?tweet .
6 }
```

Figure 7.2: SPARQL query for Example 1.

By registering this query with Twarql, every time a tweet that matches this query is streamed, the system will update the user with a new location. On the client side a user may choose to show that on a map, or create statistics of popularity of the product across the country.

Example 2: Sentiment. "Give me all people that have said negative things about my product."

Sentiment analysis is an open research problem that we do not attempt to solve in this work. Our focus is rather to demonstrate the use of sentiment annotations in the context of brand tracking. We employed a naive sentiment annotator that is based on dictionaries of positive and negative words to generate example data for this demonstration.

[3]Prefixes have been omitted for conciseness.

```
SELECT ?user
WHERE {
  ?tweet sioc:has_creator ?user .
  ?tweet moat:taggedWith dbpedia:IPad .
  ?tweet twarql:sentiment twarql:Negative .
}
```

Figure 7.3: SPARQL query for Example 2.

Example 3: Related entities. "What competitors are being mentioned with my product?"

```
SELECT ?competitor
WHERE {
  dbpedia:IPad skos:subject ?category .
  ?competitor skos:subject ?category .
  ?tweet moat:taggedWith ?competitor .
}
```

Figure 7.4: SPARQL query for Example 3.

This use case requires merging streaming data with background knowledge information (*e.g.* from DBpedia). Examples of *?category* include *category:Wi-Fi_devices* and *category:Touchscreen_portable_media_players* amongst others. As a result, without having to elicit all products of interest as keywords to filter a stream, a user is able to leverage relationships in background knowledge to more effectively narrow down the stream of tweets to a subset of interest.

Note that all four use cases focus on retrieving items that are not tweets. The information extraction and annotation from microposts enables data aggregation in different dimensions, opening numerous possibilities for analysis.

Twitter has the potential to generate many triples with user opinions, and other observations that are useful to many use cases. But as more and more triples are generated, it becomes obvious the need to control information overload. Twarql offers focused streams based on SPARQL queries as a solution to this problem.

Twarql integrates our contributions with relevant work the Social Semantic Web [John G. Breslin and Alexandre Passant and Stefan Decker, 2010] realm and in the Citizen Sensing [Sheth, 2009] area. Particularly, Twarql adds: (1) an ontology stack for representing microblogging information, built in the context of the SMOB microblogging platform; (2) sparqlPuSH [Passant and Mendes, 2010] a way to push the results of SPARQL queries matching new data loaded in the triple store; (3) Cuebee (http://cuebee.sf.net) [Mendes et al., 2008], an interface for knowledge guided query formulation.

In this case study, we focused on demonstrating how our system enables the analysis of subsets of micropost feeds through flexible and expressive querying. We discussed its application in the context of brand tracking. Twarql is also available as open-source at http://twarql.sf.net

7.3 Understanding Website Similarity

Query logs record the actual usage of search systems and their analysis has proven critical to improving search engine functionality. Yet, despite the deluge of information, query log analysis often suffers from the sparsity of the query space. Based on the observation that most queries pivot around a single entity that represents the main focus of the user's need, we propose the application of KBT to search queries in order to enable the representation of query logs through an *entity-aware click graph* [Mendes et al., 2012b]. In this representation, we decompose queries into entities and modifiers, and measure their association with pages clicked by users.

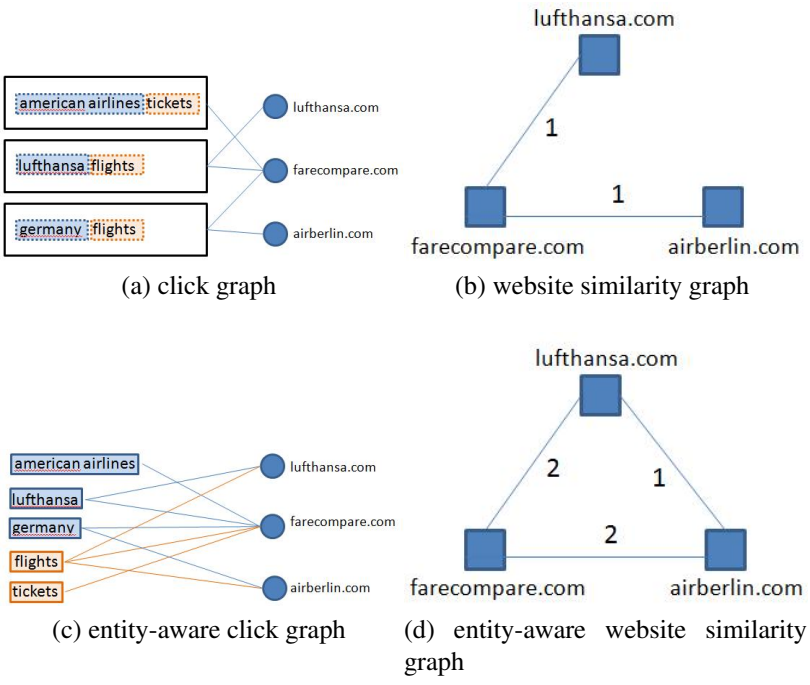
In this case study, the consumer is the search engine user, who is looking for websites about particular entities. Consumers have an active role in this case. In every interaction with a search engine, a relationship between surface form, website and user is recorded. We used the "default sense" disambiguator and resolved surface forms to their most popular interpretations. We used the information that many consumers associate an entity to a

website in order to classify websites according to the services they provide.

We evaluate the benefits of this approach on the crucial task of understanding which websites fulfill similar user needs, contrasting it with other click-based and content-based approaches.

7.3.1 Entity-aware Click Graph

A typical representation of a query log is the click graph (Figure 7.5a), a bi-partite graph where the nodes are queries $Q = \{q_1, ..., q_{|Q|}\}$ and URLs $U = \{u_1, ..., u_{|U|}\}$ or the Web sites (hosts) $S = \{s_1, ..., s_{|S|}\}$ hosting the URLs. An edge connects a query q_a to a URL u_b (or website S_c) if there has been at least one search session in the data where a user clicked on the URL (respectively, website) after issuing the query, but before issuing another query.



(a) click graph (b) website similarity graph

(c) entity-aware click graph (d) entity-aware website similarity graph

Figure 7.5: Different views on the query log by the traditional click graph 7.5a and the entity-aware click graph 7.5c. Changing the click graph model alters website similarity graph (7.5b, 7.5d).

The entity-aware click graph models relationships between entities and modifiers ap-

pearing in queries and clicked sites, and can be defined as the union of two bipartite-graphs. Let $E = \{e_1, ..., e_{|E|}\}$ be the set of all entities and $M = \{m_1, ..., m_{|M|}\}$ the set of all modifiers. We define $CG_{entity} = (E \cup S, (e_i, s_j))$ where an edge (e_i, s_i) exists if a user searched with keywords containing the entity e_i and visited site s_j. Similarly for modifiers, we define $CG_{modifier} = (M \cup S, (m_k, s_l)$ where an edge (m_k, s_l) is present if the modifier m_k was part of a query that led to a click on s_l. The entity-aware click graph is then defined as $CG = CG_{entity} \cup CG_{modifier}$. Figure 7.5c shows an example of an entity-aware click graph.

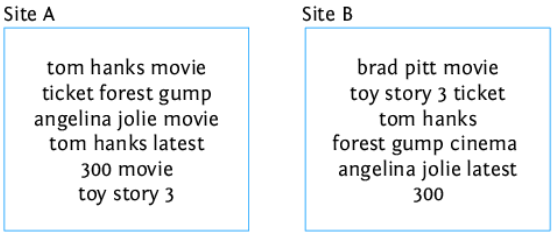


Figure 7.6: Queries leading to two different sites

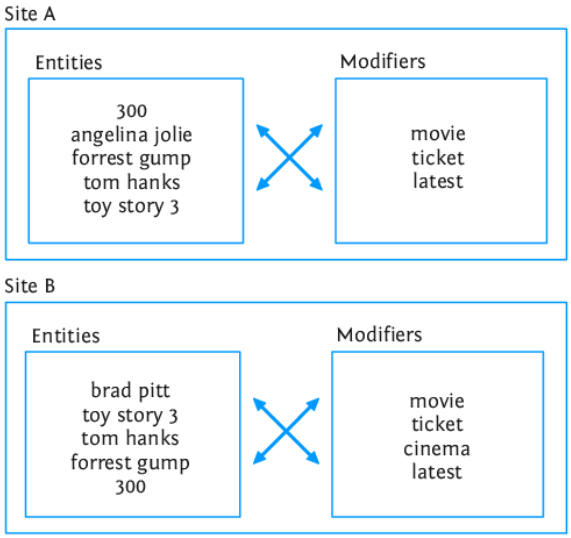


Figure 7.7: Queries from Figure 7.6 broken down into entity and modifier

7.3.2 Website Similarity Graph

Given a bipartite click-graph, computing a website similarity graph is analogous to computing the query similarity graph [Baeza-Yates and Tiberi, 2008]. This can be done, for example, by measuring the overlap of queries received by two web sites. As the example in figures 7.6 and 7.7 illustrate, website similarity graphs obtained from click graphs may fail to capture all relevant relationships. We propose to address this problem using the aforementioned entity-aware click graph, as partitioning queries reveals relationships that were obscured by the monolithic treatment of queries in the regular click graph.

7.3.3 Results

We evaluate one website similarity graph at a time. For each website s_i in S_{odp}, we collect all edges (s_i, s_j); $s_i, s_j \in S_{odp}$ from the similarity graph, and assign a score with regard to the gold standard Rel_{odp}.

$Graph$	$P@5$	$Avg(\|E\|)$
SG_{query}	0.4380	1.75
SG_{entity}	0.3140	1.94
$SG_{modifier}$	0.4500	2.01
SG_{word}	0.3840	2.26
SG_{ratio}	**0.4640**	**3.22**

Table 7.2: Precision at 5 ($P@5$) results and average number of edges returned $Avg(|E|)$ for each similarity graph.

We use P@n as defined by Deng et al. for query similarity evaluation [Deng et al., 2009] and apply it to our website similarity evaluation, so that: $P@n = \sum_{i=1}^{n} Rel_{odp}(s_i, s_j)/n$. We compute P@5 and average the values over all websites. This assesses the performance on our main intended task, i.e. suggesting a small number of similar websites to show to users in an online scenario.

Table 7.2 shows the performance of each similarity function. The SG_{ratio} graph provides a statistically significant improvement over the SG_{query} and SG_{word} baselines.

The SG_{ratio} graph is also a significant improvement over its components SG_{entity} and $SG_{modifier}$. Significance was tested with the Wilcoxon Matched-pair Signed-Ranks Test.

7.4 Automated Audio Tagging

There is an increasing amount of useful information being shared in audio format, including online courses, political debates and radio news. The BBC archive contains years-worth of cataloged audio. There is the need to organize this content and make it easy to find related programs.

However, audio is mostly opaque to current search engines. The annotation of audio files with entities from a KB can enable interlinking of content, complex queries over the collection, among other use cases.

One possible approach for such annotation is to employ Automatic Speech Recognition (ASR) tools to transcribe the audio into text, and then perform KBT over the transcripts. However, ASR is a difficult problem. Previous work has reported roughly 50% of token error on audio files from the BBC archive.

This poses new challenges for KBT systems, as NLP tools commonly require well formed text. Due to the unsupervised nature of our system, as well as the independence from a NLP-heavy pipeline, we are able to obtain competitive results for this use case simply by reusing DBpedia Spotlight's components.

The main problem with the direct application of annotation systems on audio transcripts is on the phrase recognition. Since there is a large number of transcription errors on tokens, systems are not able to find correct segments and match them to corresponding tags. However, there are enough correctly transcribed tokens to provide context for disambiguation of many of the concepts mentioned. Therefore, we used DBpedia Spotlight's components individually in an ad-hoc workflow that demonstrate our system's adaptability. We changed the standard annotation workflow by inverting the phrase recognition and con-

textual relatedness steps. The first step obtained URIs for concepts that are the most closely related to the full transcript. We then checked if those closely related concepts were recognized in the input, and rewarded those that were both related and mentioned. Based on the knowledge that the editorial staff considered as the most interesting tags the entities of types Person, Location, Organization as well as Thematic Concepts [Raimond and Lowis, 2012], we also gave those matches a higher weight. Our best weighting scheme achieved a TopN of 0.19, compared to 0.209 achieved by the much more computationally expensive state-of-the-art method by Raimond and Lowis [2012].

7.5 Educational Material - Emergency Management

Educational material used in training workshops can benefit from KBT to enrich training material with links to definitions of the terms being introduced, as well as providing links between related terms. Consider, for example, an Airport Emergency Management Training use case explored by researchers in the Mihailo Pupin Institute as part of the LOD2 Consortium [Nagy et al., 2012]. In emergency training, it is useful to analyze aspects of past events in order to build conceptual models of impacts and responses, as well as to study the efficiency of emergency plans.

> A common way within training workshops to envision emergency situations is to analyze emergencies that happened in the past on this airport or on any other airport and to determine if these emergencies were handled successfully and which lessons can be learned from them. These documents are published by different organizations such as ICAO (International Civil Aviation Organization), the European Parliament or the Civil Aviation Directorate of the Republic of Serbia. Moreover frequently reports about emergency incidents are published for example by the NFPA (National Fire Protection Association). [Nagy et al., 2012]

KBT enables a number of applications in this scenario, such as: (i) the search and interlinking of material in an emergency management catalog, (ii) allowing students to subscribe to emergency management content about a specific theme, (iii) semantic indexing of material with emergency management-specific terms, (iv) linking educational material with data on the Web.

The objective of this evaluation was to analyze the tool's accuracy different types of text for Emergency Management training, such as formal documents (an airport emergency plan - AEP), a table of contents the Convention on International Civil Aviation - ICAO, and a document with a reporting style retrieved from the National Fire Protection Association (NFPA). In this scenario, the most important annotations refer to general concepts such as the the hazard type, materials involved, as well as locations. An evaluation for this scenario was performed by sending a selection of documents through our system and judging each annotation according to the categories below [Nagy et al., 2012]:

- **Correct.** Annotated words are counted as correct if the correct DBpedia resource with the correct semantic meaning of the annotation was selected for annotation.

- **Borderline.** Annotated words are counted as borderline if a correct DBpedia resource was selected but the semantic meaning in this context is not correct.

- **Wrong.** Annotated words are counted as wrong if a similar DBpedia resource was selected (a resource that contains for example the annotated word, but together with other words it leads to a completely distinct semantic meaning).

- **Completely Wrong.** Annotated words are counted as completely wrong if the DBpedia resource does not have any relation to the annotated word. Annotations were only counted once per word, thus the percentage values refer to each annotated word not to each annotation.

Text	Correct	Borderline	Wrong	Completely Wrong
AEP (conf=0.5)	67.9%	21.4%	10.7%	0%
ICAO (conf=0.5)	66.8%	16.6%	16.6%	0%
NFPA (contextual=1)	100%	0%	0%	0%

Table 7.3: Test results for the EM training scenario.[Nagy et al., 2012]

The results are shown on Table 7.3. In the AEP text, the system included annotations of more general concepts, such as 'Definition', 'General' and 'Purpose', which were not considered interesting by the evaluators. They found out that when setting Confidence = 0.5, these more general annotations were avoided. They also noted that it generated fewer wrong annotations, but also fewer correct annotations. This is a reflection of the unavoidable trade-off between precision and recall.

In the ICAO text, the evaluators observed a more positive effect, where only aviation-related terms were annotated. Similarly, in the NFPA text, the keywords that summarize what happened were all annotated, thus making the resulting automatic tagging an interesting asset for rich snippets and summarization.

Conclusion

We presented a conceptual model of the KBT task. In this model, the Textual Content produced by a Creator is sent through a System to produce Annotations connecting the input to the KB. The resulting annotations can be sent to an Editor, that will produce Judgements of the fitness of those annotations for a particular Objective.

Through this conceptual model, we have described related work, including information extraction tasks such as Word Sense Disambiguation, Entity Linking, Named Entity Recognition, Keyphrase Extraction, Topic Indexing, providing a unified understanding of the semantic annotation literature.

Through the explicit identification of the factors that influence annotation fitness for use, the model allows determining when results from different approaches can be directly compared, and when they can be discussed in light of potential evaluation set up disadvantages. Moreover, through the definitions of difficulty to annotate provided in this model it is possible to evaluate strengths and weaknesses of 'black box' systems providing annotations as a service, such as OpenCalais, AlchemyApI, Zemanta, Yahoo! CAP, among others.

A system based on this model was developed and evaluated with widely-used corpora, showing the competitiveness and flexibility of the system. In order to further demonstrate the adaptability of our approach, we have presented a number of case studies in different domains. The selected case studies cover a variety of types of input text and requires different kinds of annotation to be produced.

The common approach for information extraction tasks is to devise rules and engineer

features to be encoded in the information extractor. Our approach is to identify ontological and linguistic features and encode them in the domain model, associated with the target knowledge base for the information extraction task at hand. This approach has farther reaching impact – beyond our own system – as potentially hundreds[1] of applications that already make use of that model can directly benefit from its extension. As a result of this work, the knowledge base has been expanded with new types of entities and millions of triples that can be directly used in online applications, without requiring to previously download and process the data through NLP pipelines.

8.1 Limitations

It has been clearly shown in the literature that specialized models can perform extremely well for specialized tasks. Powerful machine learning models are remarkably good at reproducing annotation policies, when given well engineered features and a representative sample of the kinds of annotations desired. On the one hand, a more generally adaptable solution cannot be expected to perform better than specialized solutions for each and every task. On the other hand, engineering features and carefully selecting a good sample for machine learning algorithms are both time consuming, expensive tasks that require high level of training. Adaptable solutions such as the one described in this dissertation can lower the initial costs, and serve as input for training specialized solutions.

[1] As of 04.11.2011, the main DBpedia publication has around 600 citations according to Google Scholar.

Bibliography

Eneko Agirre and Philip Edmonds, editors. *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech and Language Technology*. Springer, 1 edition, July 2006. ISBN 1402048084. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1402048084.

Alfred V. Aho and Margaret J. Corasick. Efficient string matching: an aid to bibliographic search. *Commun. ACM*, 18:333–340, June 1975. ISSN 0001-0782.

Alias-i. LingPipe 4.0.1. http://alias-i.com/lingpipe (accessed 15/04/2011), 2011.

Sophia Ananiadou. A methodology for automatic term recognition. In *Proceedings of the 15th conference on Computational linguistics - Volume 2*, COLING '94, pages 1034–1038, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/991250.991317. URL http://dx.doi.org/10.3115/991250.991317.

Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. *CoRR*, abs/1003.5699, 2010.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a Web of Open Data. *The Semantic Web*, 4825:722–735, 2007.

Sören Auer, Lorenz Bühmann, Christian Dirschl, Orri Erling, Michael Hausenblas, Robert Isele, Jens Lehmann, Michael Martin, Pablo N. Mendes, Bert Van Nuffelen, Claus Stadler, Sebastian Tramp, and Hugh Williams. Managing the life-cycle of linked data with the lod2 stack. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *International Semantic Web Conference (2)*, volume 7650 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2012. ISBN 978-3-642-35172-3. URL http://dblp.uni-trier.de/db/conf/semweb/iswc2012-2.html#AuerBDEHILMMNSTW12.

Ricardo Baeza-Yates and Alessandro Tiberi. The anatomy of a large query graph. *Journal of Physics A: Mathematical and Theoretical*, 41(22):224002, 2008. URL http://stacks.iop.org/1751-8121/41/i=22/a=224002.

Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *LREC Workshop on Linguistics Coreference*, pages 563–566, 1998.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154–165, September 2009. ISSN 1570-8268.

Jens Bleiholder and Felix Naumann. Data fusion. *ACM Comput. Surv.*, 41:1:1–1:41, January 2009. ISSN 0360-0300. doi: http://doi.acm.org/10.1145/1456650.1456651. URL http://doi.acm.org/10.1145/1456650.1456651.

Michael Buckland and Fredric Gey. The relationship between Recall and Precision. *J. Am. Soc. Inf. Sci.*, 45(1):12–19, January 1994. ISSN 0002-8231. doi: 10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;

2-L. URL http://dx.doi.org/10.1002/(SICI)1097-4571(199401)45:1%3C12::AID-ASI2%3E3.0.CO;2-L.

Razvan C. Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, 2006.

Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. Using statistics in lexical analysis. In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pages 115–164. Erlbaum, 1991.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March 1990. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=89086.89095.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, pages 249–260, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-2035-1. URL http://dl.acm.org/citation.cfm?id=2488388.2488411.

Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.

Joachim Daiber. Candidate selection and evaluation in the DBpedia Spotlight entity extraction system. Bachelor's thesis, Freie Universität Berlin, 2011. URL http://jodaiber.de/BScThesis.pdf.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *I-SEMANTICS*, pages 121–124, 2013.

Fred J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Inf. Process. Manage.*, 29(4):433–447, July 1993. ISSN 0306-4573. doi: 10.1016/0306-4573(93)90039-G. URL http://dx.doi.org/10.1016/0306-4573(93)90039-G.

Hongbo Deng, Irwin King, and Michael R. Lyu. Entropy-biased models for query representation on the click graph. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1572001.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 178–186, New York, NY, USA, 2003. ACM. ISBN 1-58113-680-3. doi: http://doi.acm.org/10.1145/775152.775178. URL http://doi.acm.org/10.1145/775152.775178.

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, March 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972450.972454.

Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, January 2007. ISSN 1041-4347. doi: 10.1109/TKDE.2007.9. URL http://dx.doi.org/10.1109/TKDE.2007.9.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Scaling wikipedia-based named entity disambiguation to arbitrary web text. In *Proceedings of the WikiAI 09 - IJCAI*

Workshop: User Contributed Knowledge and Artificial Intelligence: An Evolving Synergy, Pasadena, CA, USA, July 2009.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *In ACL*, 2005.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 668–673, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-613-0. URL http://dl.acm.org/citation.cfm?id=646307.687591.

Andrés García-Silva, Max Jakob, Pablo N. Mendes, and Christian Bizer. Multipedia: enriching dbpedia with multimedia information. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP '11, pages 137–144, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0396-5. doi: http://doi.acm.org/10.1145/1999676.1999701. URL http://doi.acm.org/10.1145/1999676.1999701.

Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/992628.992709. URL http://dx.doi.org/10.3115/992628.992709.

Daniel Gruhl, Meenakshi Nagarajan, Jan Pieper, Christine Robson, and Amit P. Sheth. Context and domain knowledge enhanced entity spotting in informal text. In *8th International Semantic Web Conference (ISWC2009)*, pages 260–276, October 2009. URL http://data.semanticweb.org/conference/iswc/2009/paper/research/174.

Ramanathan V. Guha and Rob McCool. Tap: A semantic web test-bed. *J. Web Sem.*, 1(1): 81–87, 2003.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artif. Intell.*, 194:130–150, January 2013. ISSN 0004-3702. doi: 10.1016/j.artint.2012.04.005. URL http://dx.doi.org/10.1016/j.artint.2012.04.005.

Brian Hammond, Amit Sheth, and Krzysztof Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogeneous content. In V. Kashyap and L. Shklar, editors, *Real World Semantic Web Applications*, pages 29–49. IOS Press, 2002.

Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

Joseph Hassell, Boanerges Aleman-Meza, and I. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. In Isabel Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Mike Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 44–57. Springer Berlin / Heidelberg, 2006.

Mihály Héder and Pablo N. Mendes. Round-trip semantics with sztakipedia and dbpedia spotlight. In *Proceedings of the 21st World Wide Web Conference, WWW 2012 (Companion Volume)*, pages 357–360, 2012.

S. Hellmann, C. Stadler, J. Lehmann, and S. Auer. DBpedia Live Extraction. *On the Move to Meaningful Internet Systems: OTM 2009*, pages 1209–1223, 2009.

Morten Hertzum and Erik Frøkjær. Browsing and querying in online documentation: A study of user interfaces and the interaction process. *ACM Trans. Comput.-Hum. Interact.*, 3(2):136–161, June 1996. ISSN 1073-0516. doi: 10.1145/230562.230570. URL http://doi.acm.org/10.1145/230562.230570.

Trivikram Immaneni and Krishnaprasad Thirunarayan. A unified approach to retrieving web documents and semantic web data. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *The Semantic Web: Research and Applications*, volume 4519 of *Lecture Notes in Computer Science*, pages 579–593. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72666-1. doi: 10.1007/978-3-540-72667-8_41. URL http://dx.doi.org/10.1007/978-3-540-72667-8_41.

Anja Jentzsch, Christian Bizer, and Richard Cyganiak. State of the LOD Cloud, September 2011. URL http://www4.wiwiss.fu-berlin.de/lodcloud/state/2011-09_index.html.

Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the TAC2011 Knowledge Base Population Track. In *Proceedings of the Text Analysis Conference (TAC 2011)*, 2011.

John G. Breslin and Alexandre Passant and Stefan Decker. *The Social Semantic Web*. Springer, 2010.

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Kyo Kageura and Bin Umino. Methods of automatic term recognition: a review. *Terminology*, 3(2):259–289, 1996. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.25.3740.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: http://doi.acm.org/10.1145/1557019.1557073. URL http://doi.acm.org/10.1145/1557019.1557073.

Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3c recommendation, W3C, February 1999. URL http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, to appear, 2013.

Douglas Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38:33–38, 1995.

Chin-Yew Lin and Eduard H. Hovy. The automated acquisition of topic signatures for text summarization. In *COLING*, pages 495–501, 2000.

M. E. Maron. On indexing, retrieval and the meaning of "about". *Journal of the American Society for Information Science*, 28:38–43, 1977.

Andrew McCallum, Karl Schultz, and Sameer Singh. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *NIPS*, 2009.

Paul McNamee, Hoa Trang Dang, Heather Simpson, Patrick Schone, and Stephanie M. Strassel. An evaluation of technologies for knowledge base population. In *LREC*. European Language Resources Association, 2010.

Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-63-3. URL http://dl.acm.org/citation.cfm?id=1699648.1699678.

Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 563–572, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124364. URL http://doi.acm.org/10.1145/2124295.2124364.

Pablo N. Mendes, Bobby McKnight, Amit P. Sheth, and Jessica C. Kissinger. Tcruzikb: Enabling complex queries for genomic data exploration. In *Proceedings of the 2th IEEE International Conference on Semantic Computing (ICSC 2008)*, pages 432–439, 2008.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.

Pablo N. Mendes, Max Jakob, and Christian Bizer. DBpedia for NLP: A Multilingual Cross-domain Knowledge Base. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 2012a. ISBN 978-2-9517408-7-7.

Pablo N. Mendes, Peter Mika, Hugo Zaragoza, and Roi Blanco. Measuring website similarity using an entity-aware click graph. In *21st ACM International Conference on Information and Knowledge Management (CIKM'12)*, pages 1697–1701, 2012b.

Pablo N. Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123, 2012c.

Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, New York, NY, USA,

2007. ACM. ISBN 978-1-59593-803-9. doi: 10.1145/1321440.1321475. URL http://dx.doi.org/10.1145/1321440.1321475.

G.A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38 (11):39–41, 1995.

David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: http://doi.acm.org/10.1145/1458082.1458150. URL http://doi.acm.org/10.1145/1458082.1458150.

Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997. ISBN 978-0-07-042807-2.

Mohamed Morsey, Jens Lehmann, Sören Auer, Claus Stadler, and Sebastian Hellmann. DBpedia and the Live Extraction of Structured Data from Wikipedia. *Program: electronic library and information systems*, 46:27, 2012. URL http://svn.aksw.org/papers/2011/DBpedia_Live/public.pdf.

D. Nadeau and S. Sekine. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

Helmut Nagy, Stefan Schurischuster, Bert von Nuffelen, Valentina Janev, and Jan Kucera. Lod2 deliverable 6.3.1 initial evaluation, documentation, tutorials. Technical report, LOD2 – Creating Knowledge out of Interlinked Data, EU FP7 Collaborative Project Number 257943, 2012. URL http://static.lod2.eu/Deliverables/D6.3.1_Final.pdf.

Vivi Nastase. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *EMNLP*, pages 763–772, 2008.

Roberto Navigli. Word Sense Disambiguation: A Survey. *ACM Computting Surveys*, 41 (2):1–69, 2009. ISSN 0360-0300. doi: 10.1145/1459352.1459355.

Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1099–1110, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376726. URL http://doi.acm.org/10.1145/1376616.1376726.

Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. Automatic expansion of DBpedia exploiting Wikipedia cross-language information. In *Proceedings of the 10th Extended Semantic Web Conference*, ESWC 2013, 2013.

Patrick Pantel and Dekang Lin. A statistical corpus-based term extractor. In *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, AI '01, pages 36–46, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42144-0. URL http://dl.acm.org/citation.cfm?id=647462.726284.

Alexandre Passant and Pablo N. Mendes. sparqlPuSH: Proactive notification of data updates in RDF stores using PubSubHubbub. In *Scripting for the Semantic Web Workshop (SFSW2010) at ESWC2010*, 2010.

Denilson Alves Pereira, Berthier Ribeiro-Neto, Nivio Ziviani, Alberto H. F. Laender, and Marcos André Gonçalves. A generic web-based entity resolution framework. *J. Am. Soc. Inf. Sci. Technol.*, 62(5):919–932, May 2011. ISSN 1532-2882. doi: 10.1002/asi.21518. URL http://dx.doi.org/10.1002/asi.21518.

A. Pohl. Classifying the Wikipedia Articles into the OpenCyc Taxonomy. In *WoLE'12 at ISWC'12*, 2012.

Simone Paolo Ponzetto and Michael Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 192–199, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220835.1220860. URL http://dx.doi.org/10.3115/1220835.1220860.

Yves Raimond and Chris Lowis. Automated interlinking of speech radio archives. In *Linked Data on the Web Workshop*, 2012.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1138.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534. ACL, 2011. ISBN 978-1-937284-11-4. URL http://dblp.uni-trier.de/db/conf/emnlp/emnlp2011.html#RitterCME11.

Giuseppe Rizzo and Raphaël Troncy. Nerd: A framework for unifying named entity recognition and disambiguation extraction tools. In *EACL*, pages 73–76, 2012.

Matthew Rowe. Applying semantic social graphs to disambiguate identity references. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 461–475. Springer Berlin / Heidelberg, 2009.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake Shakes Twitter Users:

Real-time Event Detection by Social Sensors. In *Proceedings of the Nineteenth International WWW Conference (WWW2010)*. ACM, 2010.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18:613–620, November 1975. ISSN 0001-0782.

Evan Sandhaus. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, 2008. URL http://catalog.ldc.upenn.edu/LDC2008T19.

C. E. Shannon. Prediction and entropy of printed english. *Bell Systems Technical Journal*, pages 50–64, 1951.

Amit Sheth. Citizen Sensing, Social Signals, and Enriching Human Experience. *IEEE Internet Computing*, 13(4):87–92, 2009.

Frank Smadja. Retrieving collocations from text: Xtract. *Comput. Linguist.*, 19(1):143–177, March 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972450.972458.

M. Stevenson and Y. Wilks. Word Sense Disambiguation. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, chapter 13, pages 249–265. Oxford University Press, 2003.

Eugenio Tacchini, Andreas Schultz, and Christian Bizer. Experiments with wikipedia cross-language data fusion. In Sören Auer, Chris Bizer, and Gunnar AAstrand Grimnes, editors, *Proc. of 5th Workshop on Scripting and Development for the Semantic Web at ESWC 2009*, volume 449 of *CEUR Workshop Proceedings ISSN 1613-0073*, June 2009. URL http://CEUR-WS.org/Vol-449/Paper3.pdf.

The Nielsen Company. Global Advertising: Consumers Trust Real Friends and Virtual Strangers the Most. *Computer World Magazine*, July 2009. URL

http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/07/pr_global-study_07709.pdf.

Krishnaprasad Thirunarayan and Trivikram Immaneni. Integrated retrieval from web of documents and data. In ZbigniewW. Ras and Agnieszka Dardzinska, editors, *Advances in Data Management*, volume 223 of *Studies in Computational Intelligence*, pages 25–48. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-02189-3. doi: 10.1007/978-3-642-02190-9_2. URL http://dx.doi.org/10.1007/978-3-642-02190-9_2.

Takashi Tomokiyo and Matthew Hurst. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment - Volume 18*, MWE '03, pages 33–40, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119282.1119287. URL http://dx.doi.org/10.3115/1119282.1119287.

Raphael Volz, Joachim Kleb, and Wolfgang Mueller. Towards ontology-based disambiguation of geographical identifiers. In *I3 Workshop at WWW*, 2007.

Michael L. Wick, Aron Culotta, Khashayar Rohanimanesh, and Andrew McCallum. An entity based model for coreference resolution. In *SDM*, pages 365–376. SIAM, 2009. URL http://dblp.uni-trier.de/db/conf/sdm/sdm2009.html#WickCRM09.

Jiaping Zheng, Luke Vilnis, Sameer Singh, Jinho D. Choi, and Andrew McCallum. Dynamic knowledge-base alignment for coreference resolution. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 153–162, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-3517.